

Ohjelmistokehittäjän työelämän osaamistarpeet

Jenny Kalliosto

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma

2018



Tekijä(t) Jenny Kalliosto	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Ohjelmistokehittäjän työelämän osaamistarpeet	Sivu- ja liitesivumäärä 39 + 11
<p>Tutkimuksen tavoitteena oli selvittää ohjelmistokehittäjän osaamistarpeita. Ohjelmistokehittäjän ura on jatkuvaa uusien asioiden oppimista ja omaksumista, sillä uusia teknologioita syntyy ja vanhat teknologiat kehittyvät jatkuvasti. Ohjelmistokehittäjän osaamistarpeet eivät kuitenkaan rajoitu teknologisten ratkaisujen hallintaan vaan projekteihin kuuluu usein myös muita taitoja, kuten menetelmäosaamista, projektihallinnallisia taitoja sekä erilaisia sosiaalisia taitoja.</p> <p>Tutkimus toteutettiin web-kyselynä, jota jaettiin työelämän verkostoitumiskanavalla, eli LinkedIn-palvelussa, noin 20:lle sitä kohtaan mielenkiintoaan osoittaneelle ohjelmistokehittäjälle, noin 40:lle oman verkostoni ja heidän verkostonsa henkilölle sekä ohjelmistokehittäjiä edustavan Talented edustusyrityksen verkoston hieman yli tuhannelle jäsenelle. Kyselyyn vastasi kaikkiaan 58 ohjelmistokehityksen parissa työskentelevää henkilöä.</p> <p>Tutkimuksen tuloksina muodostui lista työvälineistä, eli kaikesta siitä osaamisesta mitä ohjelmistokehittäjä saattaa urallaan tarvita, sekä tietoa eri työvälineiden ja metodien tärkeydestä. Hajonta ohjelmistokehityskielen kesken oli suhteellisen tasaista, mutta selkeästi tärkeän pohjan ohjelmistokehitykselle muodostaa suunnittelu erityisesti käytettävyyden kannalta, tietovarastot sekä ohjelmointi.</p>	
Asiasanat Ohjelmistokehitys, työelämän osaamistarpeet, sovelluskehityksen teknologiat ja metodit	

Sisällys

1 Johdanto	1
2 Ohjelmistokehittäjän ura on jatkuvaa opiskelua	3
2.1 Työnantajan tarpeet määrittävät työntekijän tärkeyden	5
2.2 Motivaation tärkeys ja elinikäinen oppiminen	5
3 Muita aiheeseen liittyviä tutkimuksia	7
4 Tutkimuksen esittely	12
5 Tutkimuksen tulokset	14
5.1 Teknologiset ratkaisut	17
5.2 Metodit ja toimintamallit	20
5.3 Uudet teknologiat	24
6 Pohdintaa	27
6.1 Havaintoja	28
6.2 Mahdollisuuksia	32
6.3 Oma oppimiskokemukseni	33
Lähteet	35
Liitteet	40
Liite 1. Alamutkan ja Puhakan tutkimuksen tulokset	40
Liite 2. Tutkimuskysely	41
Liite 3. Osaamistarpeiden lista avoimien vastauksien tuloksista	45
Liite 4. Teknologiset ratkaisut: Yrityksen työkalut ja niiden tärkeys.	
Monivalintakysymyksiä kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä	48
Liite 5. Metodit ja toimintamallit: Yrityksen muu osaaminen ja sen tärkeys.	
Monivalintakysymyksiä kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä	49
Liite 6. Uudet teknologiat: Yrityksen panostus uusiin teknologioihin.	
Monivalintakysymyksiä kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä	50

1 Johdanto

Tutkimukseni aiheena on vastavalmistuneen ohjelmistokehittäjän työelämävalmiudet, tarkemmin sanottuna ne taidot ja osaamiset, joita työnantajat työntekijältään kaipaavat. Aloittaessani työharjoitteluni kesällä 2016 ensimmäinen suuri haasteeni oli ymmärtää termit, joita esimieheni ripotteli keskusteluun siitä, mitä tulisin työharjoitteluni aikana työstämään. Kolmannen vieraan työkalun tai termin kohdalla aloin jo miettiä, voiko opiskelupaikka edes tarjota kaikkia niitä asioita opittavaksi ja mitä osaamista työelämä todellisuudessa vastavalmistuneelta vaatii. Esimieheni kanssa kasvokkain käytyjen sekä Google hangouts keskustelujen perusteella tutkimus on tarpeellinen ja ajankohtainen.

Tiedostan aloittaneeni työharjoitteluni suhteellisen hataralta pohjalta ajatellen, että työharjoitteluun mennään oppimaan työpaikalle – mikä pitää paikkansa myös, vaikka mielestäni taitoni olisivat voineet olla laajemmat jo työharjoittelun alkaessa. Työharjoittelu avasi näkemystäni, opintojen alkupuolella keskitytään perusteiden haltuunottoon ja vasta myöhemmin opintojen aikana selviää laajemmin mitä kaikkea ohjelmistokehittäjällä on opittavanaan. Kaikki vielä tuolloin vieraat termit synnyttivät kimmokkeen selvittää mitä kaikkea työelämässä tarvitsee, ja koska kukaan ei voi osata kaikkea, niin mihin kannattaisi keskittyä, jotta osaamistaso olisi riittävän monipuolinen ja ajantasainen.

Ohjelmistokehityksen työvälineet kehittyvät jatkuvasti, emme siis voi olettaa edes ammatillaisen hallitsevan kaikkia viimeisimpiä työkaluja tai menetelmiä, puhumattakaan oppilaitoksen mahdollisuuksista pysyä viimeisimpien trendien vauhdissa. Nopeasti kehittyvän alan opiskelijana pidänkin tärkeänä opiskelijan mahdollisuutta vaikuttaa omaan kehittymiseensä sekä sitä, että tieto tarvittavista taidoista on helposti saatavilla edes yleisellä tasolla. Tutkimukseni on tarkoitus selvittää mitä tekniikoita ja menetelmiä yritykset pitävät tällä hetkellä tärkeinä ja vaikuttaako yrityksen henkilöstömäärä tai ikä tarpeeseen eikä sen ole tarkoituksena kritisoida nykyisiä menetelmiä tai opintojen laajuuksia.

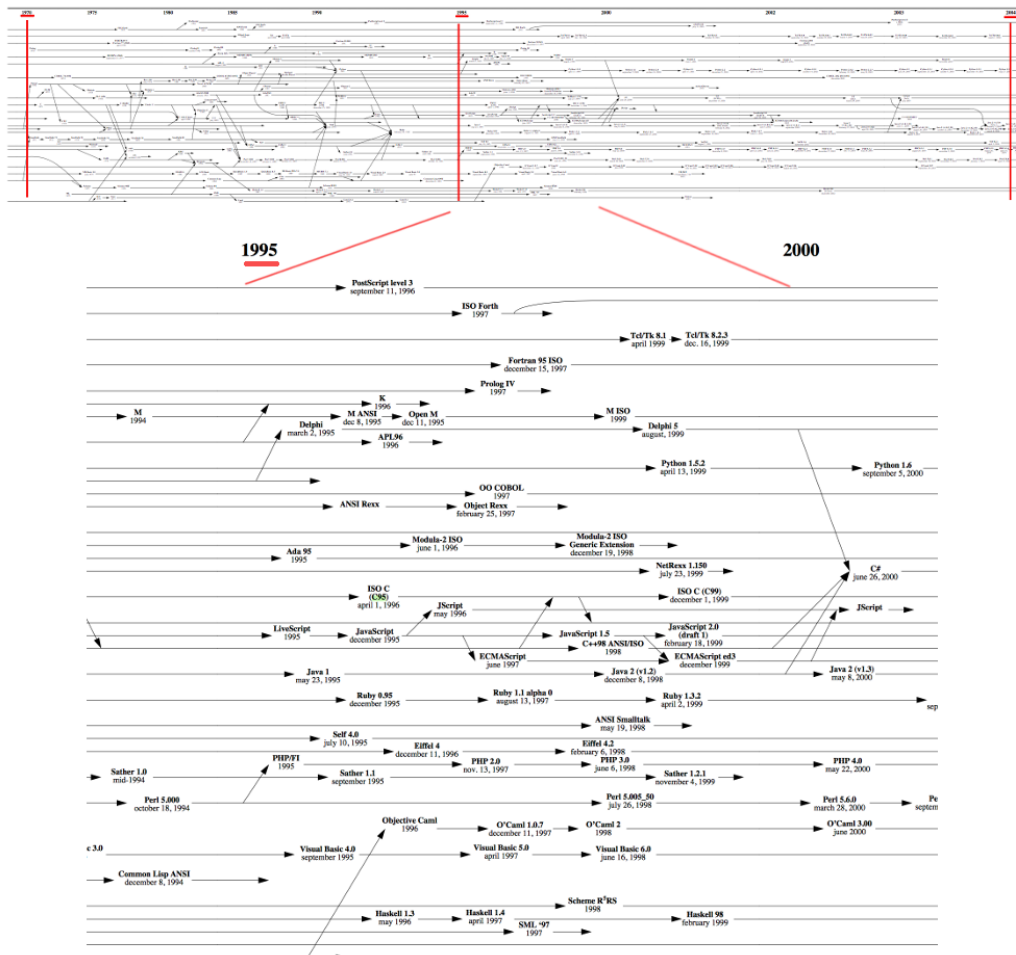
Oppilaitos voi nykymallissaan tukea opiskelijaa opintojen edetessä osin oppilaan ehdoilla, mikä tarkoittaa sitä, että opiskelijan täytyy olla aktiivinen selvittäessään mitä eri tekniikoita tai metodeja hän voisi tulevaisuudessa tarvita. Mikäli opiskelija ei tiedä tekniikoiden olemassaolosta tai itse aktiivisesti hae tietoa muualta, ei hän välttämättä ymmärrä edes kysyä niiden olemassaolon perään, eikä voida olettaa, että opettajat huomaisivat mainita mahdollisesti tulevaisuudessa tarvittavista tekniikoista, mikäli ne eivät sellaisenaan kuulu kyseisen kurssin materiaaleihin. Termistön haltuunotto yksinomaan opinto-oppaan avulla

on sen nykymuodossa työlästä, sillä opas on lähinnä suuntaa antava opintojen suhteen ja teknologiat sekä tekniikat selviävät pääsääntöisesti vasta kurssien jo käynnistyttyä. Mietinkin, voisiko opinto-opasta yksinkertaistaa tulosteettomana digiaikakautena siten, että tarvittavat teknologiat voisivat tulla tutuksi jo sitä tutkiskellessa.

Tutkimuksesta voi siis päästä hyötymään jokainen osapuoli. Opiskelijat voivat saada tietoa, mihin kannattaa panostaa, työnantajat voivat saada työntekijöitä, joiden osaamisen suuntaus on tarkempaa ja oppilaitokset voivat saada mahdollisuuden edistää opiskelijoiden laajempaa tietotaitoa entistä paremmin, edesauttaa opiskelijoiden tehokkaampaa valmistumista sekä parantaa opiskelijoiden työllistymismahdollisuuksia alalle jo opintojen aikaisemmissa vaiheissa erinäisten täytetöiden sijaan.

2 Ohjelmistokehittäjän ura on jatkuvaa opiskelua

Teknologiat ja metodit kehittyvät huimaan tahtiin, varsinkin, jos verrataan nykypäivää aikaan ennen 1990-lukua tai sen puoltaväliä, jolloin Javan ensimmäinen versio julkaistiin ja ohjelmistokehityskielen kehitys nopeutui. Osin syynä tähän saattaa olla teknologian jatkuva ja nopea kehitys sekä jo pelkästään ohjelmistokehityskielen määrä, joka on noin 2500, unohtamatta niiden vielä useampia kehyksiä. Nopeaa kehitystä kuvaa hyvin ajanjakso 1970-2004, joka esiintyy alla olevassa kuvassa (Kuva 1). Kyseessä on kuvankaappaus Lévénézin PDF julkaisusta, joka kuvaa eri ohjelmistokehityskielen kehitystä. Merkitsin kuvaan vuodet 1970, 1995 ja 2004. Ensimmäinen 25 vuotta kuvaava väli on huomattavasti lyhyempi, kuin jälkimmäinen 9 vuotta kuvaava väli. Koko PDF kuului lähteisiin ja on saatavilla <https://www.levenez.com/lang/lang.pdf>. (Kinnersley 2018, Lévénéz 2017).



Kuva 1. 34 vuoden ohjelmistokehityskielen kehitystahtia kuvaava aikajana vuosina 1970-1995 ja 1995-2004, joista aikaväli 1995-2000 korostettuna (Lévénéz 2017).

Yksinomaan käytetyn teknologian ikä ei määritä sen tarpeellisuutta ja käytettävyyttä, vaan esimerkiksi finanssi- ja pankkialalla käytössä olevaan pääsääntöisesti jo eläköityneeseen Cobol-osaajien ryhmään koulutetaan edelleen uusia osaajia, sillä isoa osaa järjestelmistä ei välttämättä tulla edes modernisoimaan (Virtanen 2018). Jo kauemmin alalla toimineet ohjelmistokehittäjät viittaavat jopa hyvin sarkastisesti aikaan, jolloin ohjelmointikieli Java kehittyi tiuhaan tahtiin. Tutkimukseni idean synnyn aikoihin keskustellessani ohjelmointikieli Javan Spring- ja Spring Boot kehyksien i'istä työharjoittelun esimieheni Koskikallion kanssa hän kertoi, että sitä aikaa ei tule ikävä, kun Javalla tehdyistä kirjastoista tuli uusia versioita viikoittain. Luonnollisesti aina on joitain teknologioita, jotka ovat isojen ja nopea-tempoisten muutosten kourissa, kuten esimerkiksi JavaScript-kirjastot, ja tulevaisuudessa jonkin muun ohjelmistokehityksen teknologia. (Koskikallio 2016).

Ohjelmistokehityksen työvälineiden jatkuvasta kehityksestä Koskikallion mainitseman JavaScriptin lisäksi toimii hyvänä esimerkkinä Spring- ja Spring Boot kehyksien vaiheet erityisesti vuosina 2011-2015. Haaga-Heliassa opiskeltiin vuonna 2016 pääsääntöisesti kahdesta aiemmin mainitsemastani kehyksestä eli tekniikasta vuonna 2002 julkaistua Springiä (Springtutorials 2015), joka oli jo silloin varsin iäkäs ohjelmistokehityksen työkalu verrattuna esimerkiksi Spring Bootiin, jonka ensimmäinen versio julkaistiin tammikuussa 2014 (Spring.io 2014). Walls kirjoitti vuonna 2011 julkaistun Spring in Action kirjansa kolmannen painoksen esipuheessa ylistystä kehittyneestä Springistä (Walls 2011, xv-xvi) ja vuonna 2015 julkaistun Spring Boot in Action -kirjan esipuheessa (Walls 2015, ix-xi) hän kertoi kuvitelleensa ennen Spring Bootia, ettei Springin voittanut tulisi, mutta toisin kävi.

Spring, Spring Boot sekä JavaScriptin kirjastot ovat esimerkkeinä kuitenkin vain pintariipaisu monista nopeasti kehittyvistä tekniikoista ja työvälineistä, joita opiskelijan odotetaan osaavan käyttää työelämäänsä siirtyessään. Ohjelmistokehittäjä joutuu kouluttautumaan uusien tekniikoiden kehittyessä käytännössä läpi koko työuransa (Koskikallio 2016). Omaamalla hyvän pohjan ohjelmistokehityksessä on uuden oppiminen sekä tekniikoiden päivittäminen helpompaa. Esimerkiksi tuntemalla MVC-arkkitehtuurin (model-view-controller suom. malli–näkymä–käsittelijä) kiitettävästi uudet metodit ovat käyttöönottaessa jo huomattavasti selvempiä (Karenko 2016). Tätä ajatusta tukee myös se, että lähes aina eri ohjelmistokehittäjien kanssa keskustellessani olen kohdannut ajattelman siitä, että jos osaa jo yhden kielen hyvin on uuden oppiminen helpompaa.

2.1 Työnantajan tarpeet määrittävät työntekijän tärkeyden

Varsinkaan pienillä yrityksillä ei yleensä ole mahdollisuutta palkata tuottamatonta työntekijää (Koskikallio 2016). Yrityksen tarkoitus on kuitenkin tuottaa voittoa, ja mikäli harjoittelija tai työntekijä ei välttämättä saavuta harjoittelun aikana taitoineen tasoa, jossa tuottoa syntyy ei häntä kannata palkata, sillä uuden työntekijän palkkaus on kallis prosessi ja työntekijä maksaa yritykselle huomattavasti enemmän kuin pelkät palkkakulut (talousverkko). Useimmiten ohjelmistokehittäjän työt ovat niin sanottuja asiantuntijatöitä. Ali-Mattilan (2015) mukaan asiantuntijayritysten suurimpiin menoeriin lukeutuvat palkat ja sosiaalikulut. Vaikka Ali-Mattilan blogi pohjautuu uusien työkalujen hankintaan, on hänellä vahva näkemys kirjoituksessaan; työntekijän aika on rahaa ja tähän viitaten työntekijän osaaminen ja yrityksen kannattavuus palkata juuri tämä henkilö useampien henkilöiden joukosta riippuu opituista tekniikoista ja taidoista. (Ali-Mattila 2015). Tästä johtuen työnantajien tarpeet, eli muun muassa teknologia- ja menetelmäosaamisen tasot, määrittelevätkin paljolti sen, mitä opiskelijan olisi kannattavaa opiskella.

Muihin aloihin verrattuna on tietotekniikan alalle kuitenkin helppo työllistyä. Varsinaista koulutusta ei välttämättä tarvita, jos tietotaito ja osaaminen ovat kohdallaan. Ohjelmistokehittäjän osaamisprofiilia ei määritellä todistusarvosanoilla, vaan sillä mitä hän oikeasti osaa, eikä ole olemassa mitään säännöstöä siitä kuka voisi toimia asiantuntijana alalla. (Jäntti 2012.)

2.2 Motivaation tärkeys ja elinikäinen oppiminen

Euroopan parlamentti suosittelee sisällyttämään ICT- ja digitaaliset taidot elinikäisen oppimisen piiriin (Lex-Europa Recommendation 2006/962/EC). Saavuttaakseen ja ylläpitääkseen työnantajan toivoman tason on opiskelijan syytä jatkaa elinikäistä opintomatkaansa edelleen valmistumisensa jälkeen työelämään päästyäänkin.

Valtaosa ohjelmistokehitys-yrityksistä järjestää työntekijöilleen lisäkoulutusta, sillä ei voida edes olettaa työntekijän osaavan aina kaikkea mitä hänen tarvitsee osata. Mikäli kouluttaminen nähdään oleellisena tai tarpeellisena, sitä pääsääntöisesti järjestetään. Koulutus ei välttämättä tapahdu kuitenkaan varsinaisena koulutuksena, vaan se voi tapahtua myös työnteon ohella ohjaamisena ja neuvomisena tai seminaareina. Työnkuvan muuttuessa joko täysin tai osittain yleensä järjestetään tarveperäistä koulutusta, joka mitoitetaan tarpeen mukaan (Sipria 2018).

Osassa yrityksistä saatetaan järjestää koulutusta jopa tarveperäisyyden rajojen yli esimerkiksi budjetoimalla työn ohkeen koulutuspäiviä vakituisesti ja työntekijän kiinnostuksen kohteiden mukaisesti. Työntekijä voi erikseen sopia esimiehensä kanssa, missä tai miten hän haluaisi kouluttautua ja jakaa sitten kokemuksensa koulutuksen jälkeen kollegoidensa kanssa (Hakkarainen 2018.)

Yksi hyvä tapa oppia, on opettaa muille jotain mitä jo osaa, esimerkiksi mestari ja kisällitoiminnalla, jolla tarkoitetaan yksinkertaistetusti sitä, että työparin kokeneempi ohjelmistokehittäjä opettaa ja vahvistaa vähemmän kokeneen osaamista työn ohessa. Toisinaan koulutustapahtumat voivat olla jopa enemmänkin sosiaalisia ja viihteellisiä tapahtumia, joissa ihmiset pääsevät opettamaan joko työnsä tai vapaa-aikansa puolesta itseään kiinnostavia asioita muille ja oppimaan muilta jotain, mikä heitä kiinnostaa. Sosiaalinen toiminta ja yhteisöllisyys edesauttaa oppimista yhtä lailla kuin motivaatio ja kiinnostus opittavaa aihetta kohtaan. Ketään ei oikeastaan voi pakottaa tekemään tai oppimaan jotain, mikä ei vain kiinnosta, vaikka jokaista kannustetaankin oppimaan omien rajojensa yli. On erityisen tärkeää, ettei pelkää lähteä kokeilemaan ja sitä kautta oppimaan jotain uutta (Peurala 2018.)

Tietenkin lisäkouluttautuminen vaatii myös oma-aloitteisuutta ohjelmistokehittäjältä sekä vahvaa kiinnostusta alaan. Monet isommat yritykset järjestävät ilmaisia koulutuspäiviä ja moni ohjelmistokehittäjä hyödyntääkin niitä sekä oppimis- että verkostoitumisväylinä. Uusia asioita voi kouluttautumisen lisäksi oppia töitä tekemällä, mikäli projektin budjetti ja aika antaa myöten, voi sitä vaikka lähteä toteuttamaan jollain itselle vielä täysin vieraalla tavalla tai kielellä (Toivanen 2018.)

Hektisesti kehittyvän alan tarpeiden oppiminen ja sen tiedostaminen, mitä työnantaja tulevalta työntekijältään odottaa voi usein olla monen mutkan takana. Motivaation kehitys ja ylläpito onkin yksi ydinkysymyksistä ohjelmistokehittäjän uralla teknologioiden jatkuvan kehityksen keskellä. Aito kiinnostus ja motivoituneisuus ovat avainasemassa itsensä kehittämisessä ja uuden oppimisessa ja sisäistämisessä. Suuri osa saavutettavista taidoista vaatii opiskelijalta ja työntekijältä motivaatiota, huolimatta siitä miten hän osaamistaan kartuttaa. Pääasia kuitenkin on, toivotun ja odotetun tieto- ja taitotason saavuttaminen tai ylläpito, siten että osaamistaso vastaa työnantajan toivomuksia, puuttumatta siihen, mitä työnantaja varsinaisesti kokee riittävänä osaamisena. Motivaatio, eli halu ja into tehdä jotain voi auttaa opiskelijaa saavuttamaan toivotun tietotason. (Ryan & Desi 2000.)

3 Muita aiheeseen liittyviä tutkimuksia

Siitä huolimatta, että käytetyt teknologiat kehittyvät ja vaihtuvat suhteellisen nopealla tahdilla on ohjelmistokehittäjän taidoista toteutettu erinäisiä tutkimuksia. Pääpiirteittäin tutkimukset ovat kehittyneet laajempiin osa-alueisiin kuten ohjelmistokehityskieliin yleisesti erillisten ja erilaisten ohjelmistokehityskielen sijaan. Olen valinnut tutkimukseni vertailukohteiksi kolme eri-ikäistä tutkimusta, näistä kaksi arvottaa teknologian ja muun osaamisen tärkeysjärjestystä yhtenä massana ja yksi jaotellen edellä mainitut kahteen osa-alueeseen, kuten tämä tutkimus. Tutkimuksien erot tekevät suorasta vertailusta lähes mahdollonta, mutta samankaltaisuudet kuitenkin mahdollistavat osittaisen vertailun pääpiirteittäin.

Vuonna 2009 Tampereen teknillinen yliopisto teetti tutkimuksen, joka pohjasi Kanadalaiseen Ottawan yliopistolla toteutettuun tutkimukseen. Se selvitti 212 Suomalaisen ohjelmistokehitysalan ammattilaisen vastausten perusteella mihin tärkeysjärjestykseen tutkimukseen valitut 72 aihetta sijoittuivat. Tulokset eivät sinänsä yllätä, sillä kärkisijoilla on asioita, joita voisi pitää oman kokemuksenikin mukaisesti ohjelmistokehityksen kivijalkana; määrittelemätön ohjelmointikieli, perustasoiset ohjelmointitaidot, tietokannat ja tietorakenteet. Perusolemukseltaan tutkimus oli paljolti samankaltainen, kuin tämän tutkimuksen on tarkoitus olla ja pidän sen tuloksia (liite 1) mielenkiintoisina vertailukohteina tutkimusten noin kymmenen vuoden ikäeron vuoksi. Poikkeus samankaltaisuuteen on siinä, että yleisten termien, kuten ohjelmointikieli, sijaan tutkimukseni menee syvemmälle yksityiskohtiin ja perehtyy esimerkiksi Javan tai JavaScriptin tarpeellisuusarvioon tällä hetkellä. Teknologioiden ja metodien arvottaminen tärkeinä tai vähemmän tärkeinä on samankaltainen molemmissa tutkimuksissa. (Ala-Mutka, Puhakka 2009, 3-4.)

Ala-Mutkan ja Puhakan tutkimuksen lähteenä toiminut Kanadalainen tutkimus toteutettiin vuonna 1999. Se selvitti koulutuksen merkitystä ohjelmistokehittäjälle. Yksi osa tutkimusta käsitteli sitä, mitä vastaajat pitivät tärkeinä ja mitä vastaajien mielestä jokaisen ohjelmistokehittäjän olisi hyvä osata. On selkeästi havaittavissa, että Ala-Mutkan ja Puhakan tutkimus nojasi äärimmäisen vahvasti tähän Lethbridgen tutkimukseen, sillä tutkimuksen tuloksien jaottelu on samankaltainen molemmissa. Perehdyn kuitenkin vain niihin tuloksiin, jotka viittaavat tarpeellisuuteen työelämässä ja ainakin osa tuloksista on edelleen relevantteja, kuten seuraavasta taulukosta (taulukko 1) voi havaita. (Lethbridge 1999.)

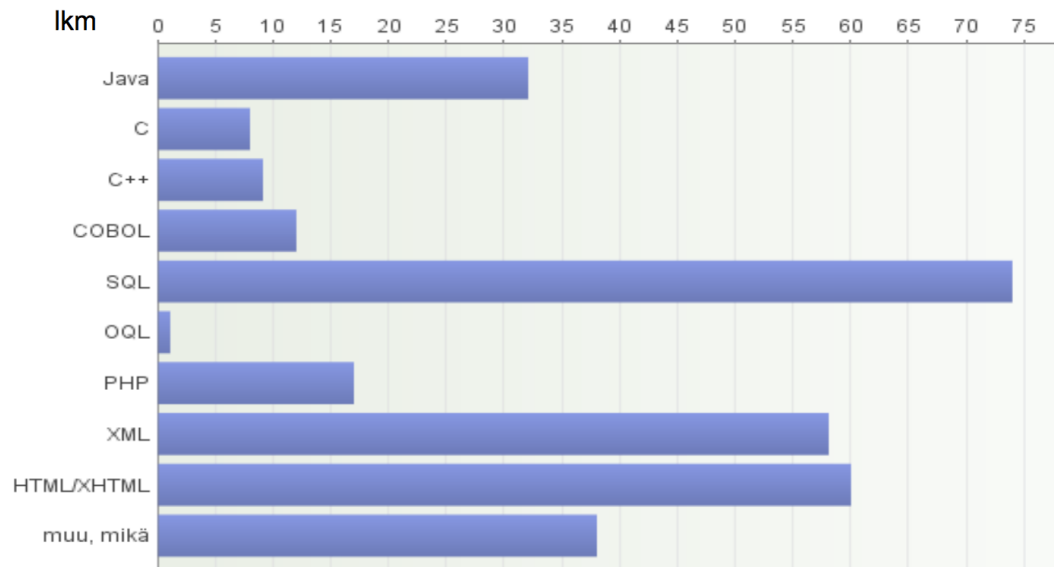
Rank	Topic	Overall Importance	Details (Q3)	Influence (Q4)
1	Specific Programming Languages	3.8	4.1	3.5
2	Data Structures	3.6	3.7	3.4
3	Software Design and Patterns	3.5	3.6	3.5
4	Software Architecture	3.4	3.5	3.3
5	Requirements Gathering & Analysis	3.4	3.5	3.3
6	HCI / User Interfaces	3.3	3.3	3.3
7	Object Oriented Concepts & Tech.	3.3	3.3	3.3
8	Ethics and Professionalism	3.3	3.2	3.4
9	Analysis and Design Methods	3.3	3.3	3.3
10	Giving Presentations to an Audience	3.3	3.5	3.1
11	Project Management	3.3	3.4	3.2
12	Testing, Verif. & Quality Assurance	3.2	3.3	3.1
13	Design of Algorithms	3.2	3.3	3.1
14	Technical Writing	3.1	3.4	2.9
15	Operating Systems	3.1	3.3	3.0
16	Databases	3.1	3.3	2.8
17	Leadership	3.0	3.1	3.0
18	Configuration and Release Management	3.0	3.3	2.8
19	Data Transmission and Networks	3.0	3.1	2.8
20	Management	2.9	2.9	2.9
21	File Management	2.8	3.2	2.4
22	Software Reliability & Fault Tolerance	2.8	2.9	2.7
23	Systems Programming	2.8	2.9	2.7
24	Network Architecture & Data Trans.	2.8	2.8	2.7
25	Negotiation	2.8	2.9	2.6

Taulukko 1. Lethbridgen tutkimuksen 25 tärkeintä aihetta niiden tärkeyden ja vaikutuksen perusteella (Lethbridge 1999, 33).

Lethbridgen sekä Ala-Mutkan ja Puhakan tutkimuksien erot liittyvät pääsääntöisesti käytettyihin teknologioihin. Sekä tutkimustapa että kysymykset ovat olleet samanlaiset tai vähintäänkin hyvin samankaltaiset. Molemmissa tutkimuksissa esitettiin samat neljä kysymystä valittujen termien vertailussa. Kysymykset koskivat sitä, kuinka paljon vastaaja oli oppinut aiheesta opintojensa aikana ja kuinka paljon hän tiesi aiheesta kyselyyn vastatesaan. Lisäksi kysyttiin, kuinka paljon tietotaidosta kysytyjen aiheiden mukaan oli ollut hyötyä työelämässä ja sitä, että kuinka paljon aiheesta oppiminen oli vaikuttanut vastaajan ajattelutapaan. Kuten aikaisemmin mainitsin, keskityn näiden tutkimuksien tuloksista niihin vastauksiin, jotka viittaavat siihen, kuinka hyödylliseksi osaaminen on osoittautunut.

Ala-Mutkan ja Puhakan tutkimusta lähteenä käyttäneessä vuonna 2012 Haaga-Helian opinnäytetyötutkimuksessa selvitettiin alumnien kokemusta siitä, miten opinnot ovat tukenet työtehtäviä ja miten opintoja voisi mahdollisesti kehittää tukemaan enemmän työelämän tarpeita. Valtaosa tutkimukseen vastanneista oli tradenomiopintojensa aikana suuntautunut ohjelmistokehitykseen ja noin puolet heistä työskenteli ohjelmistokehittäjän työtehtävissä vastaushetkellään. Hyödyllisimpinä koulutuksesta saatuina tietoina pidettiin

ohjelmointia sekä tietokantoja, joita pidettiin myös vastaajien mielestä tärkeimpinä osa-alueina. Tuloksista poimintana lähes 75 vastaajaa 107:stä oli nimennyt SQL:n tarpeelliseksi ja HTML sekä XML olivat olleet vastaajille myös tärkeitä (kuvio 1). Muina tärkeinä taitoina vastaajat olivat pitäneet muun muassa dokumentointia sekä organisaatiotaitoja.



Kuvio 1. Mitä ohjelmointi- tai määrittelykieliä vastaajat olivat tarvinneet työelämässään (Teerimäki 2012, 28).

Teerimäen tutkimuksessa keskityttiin siihen, mitä alumnit olivat opintojensa aikana oppineet sekä miten tärkeiksi he olivat oppineensa asiat kokeneet työelämänsä kannalta. Tutkittaessa vastaajien, jotka olivat suuntautumisopintojensa mukaisessa työssä, mielipidettä siitä, miten hyvin heidän koulutuksensa vastasi heidän sen hetkisiä työtehtäviään, valtaosa kaikista vastaajista koki opintojensa tukeneen työtehtäviä joko enimmäkseen tai osittain. (Teerimäki 2012, 1, 26-30, 41.)

Yhtenä haasteena tutkimuksien eroja ja samankaltaisuuksia verrattaessa on tutkimuksien ikäero ja se, mikä todellisuudessa tutkimuksen julkaisun aikaan oli tärkeää. Alla olevaan taulukkoon (taulukko 2), jonka olen pohjannut enimmäkseen Ala-Mutkan ja Puhakan sekä Lethrbridgen tutkimuksiin, olen kerännyt osasta tutkimuksissa käytetyistä termeistä vertailutaulukon. Valinnan olen tehnyt tärkeyden perusteella ja sen perusteella, minkä itse näen oleellisena omaa tutkimustani ajatellen. Jouduin hieman käyttämään omaa tulkintakykyäni siinä, mikä asia tutkimuksen listauksessa olisi lähinnä termiä, joka on hieman eriniminen tai erilainen toisessa tutkimuksessa. Osaa asioista ei ollut tarjolla kaikissa kolmessa

tutkimuksessa. Lisäksi Teerimäen tutkimuksen tuloksista on lähes mahdoton arvottaa kaikkia termejä yhtenä osana, vaan kirjasin tulokset A ja B muotoisina. A ryhmään kuulivat teknologiset ratkaisut ja B ryhmään muu osaaminen. Lisäksi, Teerimäen tutkimuksen pohjalta on mahdotonta verrata yksittäisen ohjelmistokehityskielen tärkeyttä ohjelmistokehityskieleen yleisesti, sillä kuten tässä tutkimuksessa, on Teerimäen tutkimuksessa myös enemmän tai vähemmän hajontaa eri kielten välillä.

	Lethbridge, 1999	Ala-Mutka & Puhakka, 2009	Teerimäki, 2012
Ohjelmoinnin perustaidot	-	1	A 2
Ohjelmistokehityskieli	1	2	N / A
Tietorakenne / tietoverkot	2	10	A 8
Määrittely ja suunnittelu	3	24	A 10
Sovellusarkkitehtuuri	4	20	A 11
Vaatimusmäärittely	5	5	A 7
Käytettävyys	6	12	-
Esiintymistaidot	10	9	B 3
Projektinhallinta	11	11	A 4
Testaus ja/tai laadunvarmistus	12	8	A 6
Käyttöjärjestelmät	15	17	A 5 **
Tietokannat	16	3	A 1
Dokumentointi	21 *	4	B 1
Kokous ja neuvottelutaidot	25	9	B 4
Tietoturva	42	21	A 9
Internetpalvelut, www-sovellukset	-	16	A 3

* File Management, ** Käyttöliittymät

Taulukko 2. Eroja ja yhtäläisyyksiä Lethbridgen, Ala-Mutkan ja Puhakan sekä Teerimäen tutkimuksissa (Ala-Mutka & Puhakka 2009, Lethbridge 1999, Teerimäki 2012).

Pääpiirteittäin tutkimuksien tärkeimmät osa-alueet ovat hyvin samankaltaiset. Kaikissa tutkimuksissa ohjelmoinnin perustaidot tai jokin ohjelmistokehityskieli ovat tärkeysasteikolla kärkikaksikossa. Lisäksi on havaittavissa tietokantojen tärkeyttä, sillä Ala-Mutkan ja Puhakan sekä Teerimäen tutkimuksissa tietokannat ovat yhtä lailla tärkeimmissä osaamisalueissa. Lethbridgen tutkimuksessa tietokannat sijoittuvat kuitenkin vasta 16 sijalle, mikä on ehkä selitettävissä tietokantojen kehityksen kautta, sillä tutkimus oli toteutettu vain joitain vuosia suosituksen ja nykyään yleisesti paljon käytetyn MySQL:n julkaisun 1995 (Widenius 2018) jälkeen.

Näiden aikaisemmin toteutettujen tutkimuksien tuloksia tarkastellessa on havaittavissa joitain trendejä. Esiintymistaidot sekä kokous- ja neuvottelutaidot ovat yhä tarpeellisemmassa roolissa, mitä lähemmäs nykypäivää tullaan. Tästä voin päätellä, että perinteinen introverttihenkinen sovelluskehitys on jäämässä entistä enemmän historiaan ja kehittäjiä pikemminkin kannustetaan yhteistyöhön ja oman kehitystyönsä esittelyyn. Päätelmäni vaikuttaa myös kokemuksen opintojen aikana, jolloin ohjelmistokehittäjät ovat itse

esitelleet asiakkaalle tuotostaan erillisen myyjän tai vastaavan henkilön sijaan. Esiintymistaitojen lisäksi myös internetsovellusten, tietoturvan, dokumentaation sekä testauksen suhteen on havaittavissa trendejä, jotka kuvaavat näiden osaamisalueiden tärkeyden kehitystä.

Tutkimuksien samankaltaisuuksien vuoksi uskonkin tutkimustulosten yhtenevyyteen ja oletan, etteivät yleiset tarpeet ole olennaisesti muuttuneet. Tutkimukseni kuitenkin pureutuu näitä aikaisempia tutkimuksia syvemmälle osaamistarpeiden parissa ja selvittää mitkä ovat tämän hetken ohjelmistokehittäjälle tärkeimpiä ajankohtaisia teknologioita ja metodeita.

4 Tutkimuksen esittely

Tutkimuksen tavoitteena oli selvittää minkälaista osaamista työnantajat tarvitsevat vasta valmistuneelta ohjelmistokehittäjältä. Tutkimuksessa osaamistarpeet oli jaettu kahteen osa-alueeseen; työkaluihin eli muun muassa ohjelmistokehityksen eri kielet, teknologiset ratkaisut ja hankittu osaaminen sekä muuhun osaamiseen kuten muun muassa koulutus-, myynti- ja projektihallinnalliset taidot. Työkalujen ja osaamisen lisäksi kysyttiin yrityksen suhtautumista uudempiin teknologioihin kuten viime aikoina paljon huomiota saaneeseen esineiden internetiin, eli IoT:iin. IoT verkosto on kasvanut vuosien saatossa, vuonna 2012 laitteita oli arviolta noin 8,7 biljoonaa (Soderbery 2013) ja vuonna 2017 tammikuussa laitteiden määrän uskottiin nousevan 20 biljoonaan sen vuoden aikana (Brown 2017).

Tutkimus suoritettiin web-kyselytutkimuksena. Kysely ei noudattanut klassisen tutkimuksen aineistonkeruutapaa. Vastaajien hankinnassa hyödynnettiin LinkedIn:iä, joka on työelämän verkostoitumiskanava, sekä Talentedia, joka edustaa kokeneita ohjelmistokehittäjiä ja suunnittelijoita, etsien yhteistyö-yrityksiensä projekteille tekijöitä verkostostaan. Kysely oli avoimessa levityksessä Talentedin verkoston Slack-kanavalla, jolla on hieman yli tuhat seuraajaa. Slack on erityisesti ohjelmistokehittäjien keskuudessa suosittu kommunikatiiväylä. LinkedIn:ssä kyselyä jaettiin henkilökohtaisesti kontakteilleni, heidän kontakteilleen, joita oli yhteensä noin 40, sekä kyselyä kohtaan yleistä mielenkiintoa osoittaneille noin kahdellekymmenelle henkilölle julkaistua siellä pyynnön tutkimukseen osallistumisesta. LinkedIn:ssä tapahtunut rajausta oli tarkka virhemarginaalin minimoimiseksi, eli sillä tavoittelin sitä, että vastaajat työskentelisivät pääsääntöisesti vain ohjelmistokehityksen parissa eikä esimerkiksi ruoanvalmistusalan tehtävissä. Kysely rakennettiin siten, ettei yksittäistä vastaajaa voi tunnistaa vastausten perusteella, ja siten on myös mahdoton sanoa mitä väylää pitkin vastaaja on kyselyyn törmännyt.

Tutkimuksen varsinainen valmistelu aloitettiin keväällä 2017, jolloin loin tutkimuskyselyn, joka on tutkimuksen liitteenä (liite 2). Alkaessani suunnittelemaan sitä mietin, että miten voisin saada mahdollisimman kattavaa ja hyvää dataa tuloksia ajatellen. Mikäli antaisin vastaajille vain valmiita kysymyksiä eri termeistä, voisi moni asia, josta en välttämättä edes tietäisi mitään voinut jäädä täysin huomiotta. Mikäli taas keräisin vain avointa dataa, voisivat tulokset olla jopa kaoottiset ja yhtä lailla, moni mielenkiintoinen ja tärkeä seikka voisi jäädä huomiotta. Tästä dilemmasta johtuen päätin toteuttaa kyselyn huomioiden molemmat ongelmat ja aloitin sen avoimella kysymyksellä, jotta vastaaja pääsisi ensin miettimään mitä hänelle tulee ensimmäisenä mieleen, kun kysytään tärkeimpiä työvälineitä ja

vasta sen jälkeen hän pääsisi vastaamaan monivalintakysymykseen, jossa hän pääsisi arvottamaan eri työvälineiden tärkeyttä.

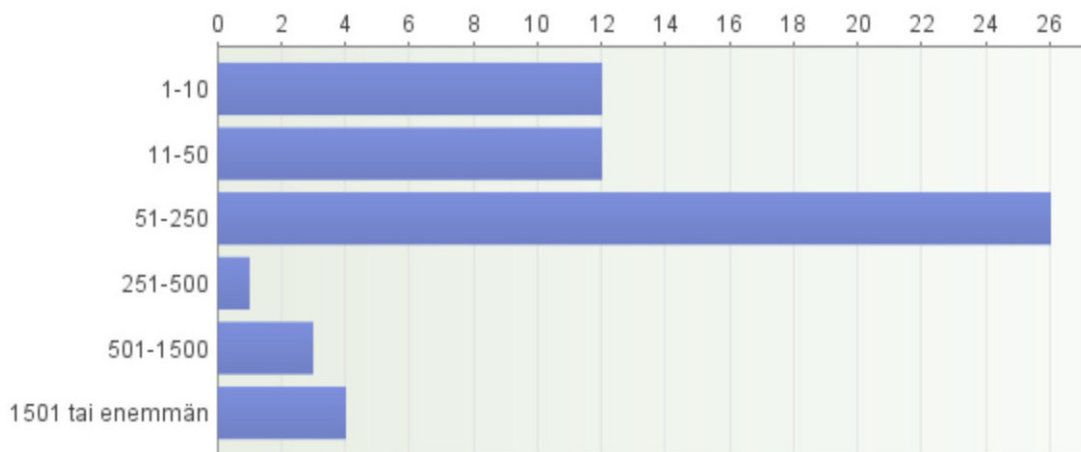
Monivalintakysymyksen termejä valitessani pyrin siihen, että tarjolla olisi monenlaisia ja monen ikäisiä vaihtoehtoja, mutta kuitenkin siten, ettei vaihtoehtoja olisi liikaa. Esimerkiksi teknologisiin ratkaisuihin valitsin ohjelmistokehityskieliä, joiden oletin olevan nyt tai olleen joskus suhteellisen suosittuja tai muutoin maineikkaita kuten Javan, C#:n ja Cobolin. Vajeellisten monivalintakysymyksien listan lisäksi vastaajilla oli mahdollisuus valita myös ”jokin muu”, jolloin he saivat itse nimetä käyttämänsä teknologian tai metodin, joka ei muutoin löytynyt listaltani.

Kyselyn tyylillä tavoittelin siis laajaa tietämystä eri teknologioista ja metodeista, en yksinomaan pääsuuntauksista kuten ohjelmistokehitys ja tietokannat vaan käytetyistä kielistä tai jopa sovelluksista, joita ohjelmistokehittäjät käyttävät. Omakohtaisena kokemuksenani sovellukset, joita käytetään, riippuvat paljolti yrityksestä ja jopa siitä millä ohjelmistokehityskielellä sovellusta käytetään. Sovellukset ovat jokseenkin irrelevantteja tutkimusta ajatellen, sillä usein on makuasia, mitä sovellusta käytetään mutta pidän mahdollista tietoa sovelluksista mielenkiintoisena, joskaan en niin oleellisena, että lähtisin niitä erityisesti monivalintakysymyksissäni selvittämään.

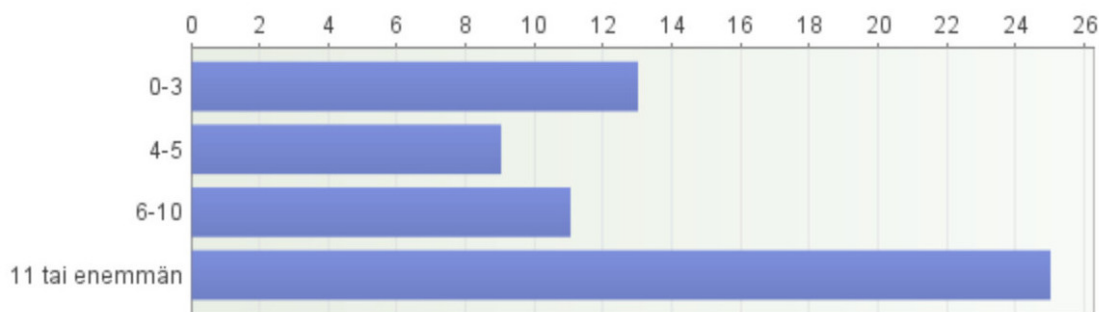
Kysely julkaistiin vasta lokakuun 2017 lopulla. Syy myöhäisempään julkaisuajankohtaan oli lomakausi ja lomaltapaluuajan oletettu kiireellisyys. Lokakuun lopulta marraskuun lopulle on pääsääntöisesti melko kiireetöntä aikaa, jolloin oletin vastaajilla olevan paremmin aikaa paneutua kyselyyn. Uskon, että moni yrityksistä saa paljon vastaavanlaisia kyselypyyntöjä, joten pyrin panostamaan kyselyn laatuun sen määrän sijaan ja muutaman testivastauksen perusteella kyselyn vastaamiseen meni keskimäärin 7 minuuttia vastauksen laajuudesta riippuen. LinkedIn-palvelussa kyselyä jaettiin koko kyselyn aukiolon ajan siitä kiinnostuneille, Talentedin slack-kanavalla se oli vapaassa levityksessä joulukuun ajan. Kysely oli avoimena kaikkiaan noin kaksi kuukautta lokakuun loppupuolelta vuodenvaihteeseen.

5 Tutkimuksen tulokset

Vastaajia kyselyssä oli yhteensä 58. Valtaosa, eli 50, vastaajaa toimi vastaushetkellään enintään 250 henkilöä työllistävässä (kuvio 2) yrityksissä ja yli puolet, eli 36, yrityksistä oli perustettu vähintään 6 vuotta sitten (kuvio 3). Kahden yrityksen suurimman tai ensimmäisenä perustetun toimipisteen maantieteellinen sijainti oli ulkomailla ja 44 yrityksistä sijaitsi pääkaupunkiseudulla. Suurin osa vastaajista toimi ohjelmistokehittäjinä tai suunnittelijoina, vastaajien joukkoon mahtui kuitenkin myös muun muassa johtajia.



Kuvio 2: Kuinka monta henkilöä vastaajan yritys työllistää.

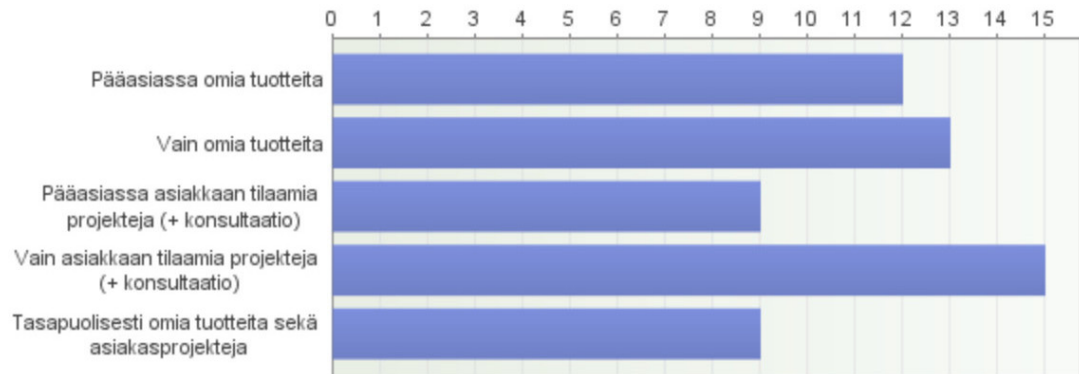


Kuvio 3: Kuinka kauan yritys on ollut toiminnassa.

Yrityksien koko ja ikäeroilla on havaittavissa lievää yhteyttä. Kaikkiaan suurin osa yrityksistä oli 51-250 henkilöä työllistäviä ja ainoastaan yli kymmenvuotiaiden yritysten joukosta löytyi henkilöstömäärältään tätä suurempia yrityksiä.

Ohjelmistokehitystaloilla on eri toimintamalleja, yritykset voivat tuottaa täysin omia sovelluksia tai toimia esimerkiksi konsultaatioyrityksinä, jolloin yrityksen työntekijä tekee

työpanoksensa toiselle yritykselle, ehkä jopa toisen yrityksen tiloissa, vaikka virallisesti on oman työpaikkansa palkkalistoilla. Tutkimukseen vastanneiden henkilöiden työpaikkojen toimintamallit jakautuivat suhteellisen tasaisesti konsultaation, omien tuotteiden ja näiden välimuotojen kesken (kuvio 4).



Kuvio 4: Toteuttaako ja myykö yritys pääsääntöisesti omia sovelluksia vai onko yrityksen toiminta tilaus- tai konsultaatiopalvelua?

Kokonaisuutena vastaajien suurin mielenkiinto kohdistui erityisesti ohjelmistokehityskieliin, tietokantaratkaisuihin, käyttäjäystävällisyyteen sekä pilvipalveluihin. Ohjelmistokehityskielen määrä aikaansai melkoisesti hajontaa eri kielten kesken, mutta siitä huolimatta tämän tutkimuksen vastausten perusteella tärkeimpänä ohjelmistokehityskielenä pidetään Javaa. Tietokantoja, käyttäjäystävällisyyttä sekä pilvipalveluita lähes kaikki vastaajat pitivät erittäin tärkeinä.

Koska tutkimus toteutti niin kutsuttua mixed-methods menetelmää, sen tuloksista pystyi keräämään niin kvalitatiivista eli laadullista, kuin kvantitatiivista eli määrällistä dataa. Kvalitatiivisina tuloksina avulla tutkimus tuotti listan (liite 3) eri työvälineistä, eli eri teknologioista, metodeista ja osaamisesta mitä vastaajat työssään tarvitsevat. Kvantitatiivisina tuloksina saatiin tietoa siitä, kuinka iso osa vastaajista kokee tietyn työvälineen oleellisena omassa työssään.

Avoimien kysymysten vastaukset antoivat aika hyvin kuvaa siitä, minkä tyyppisiä ratkaisuja yrityksillä kaiken kaikkiaan on yleisesti käytössä. Laajoin vastaus, jonka sain avoimiin vastauksiin päätettiin sanoa: ”tämä on käsittämättömän laaja kysymys” ja ymmärrän täysin vastaajan mielipiteen. Ilman eri jaottelua kaiken käytetyn osaamisen listaaminen saattoi olla tunnolliselle vastaajalle urakka, sillä kyselyssä selvittämiini työkaluihin kuuluvat niin

fyysiset sovellukset, kielet ja muut teknologiat kuin käytännössä aineettomammat projektihallinnalliset taidot, toimintatavat, käyttöliittymäsuunnittelu ja muut ohjelmistokehittäjän taidot. Avoimen vastauksen antoi 57 vastaajaa 58:sta.

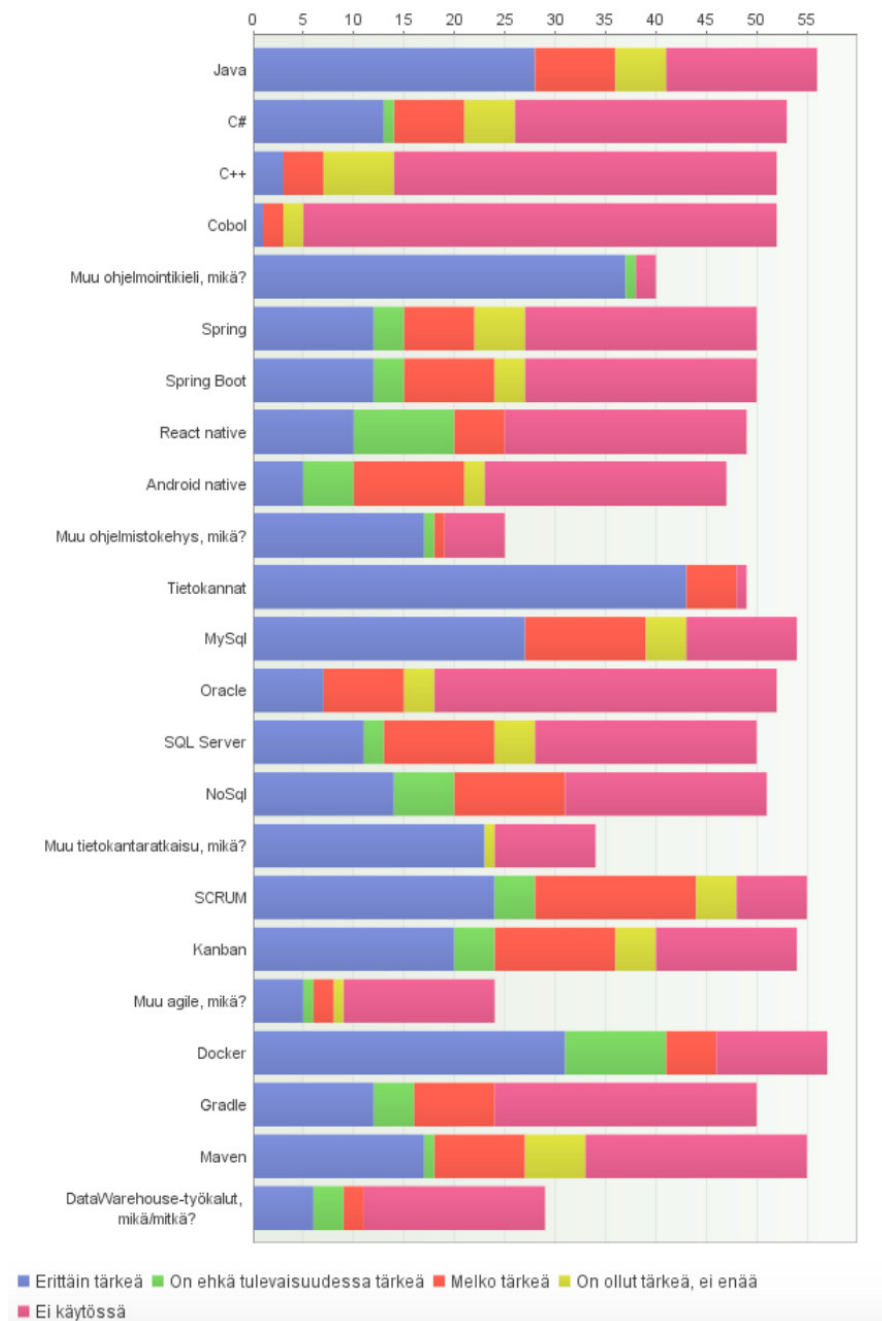
Käytettyjen teknologioiden tarve pohjautuu paljolti sille, mitä tuotteita yritykset tuottavat tai minkälaisia palveluita yritys tarjoaa. Se toteuttaako yritys enemmän asiakaslähtöisiä projekteja vai omia sovellusprojekteja saattaa osin vaikuttaa siihen minkä tyyppisiä ratkaisuja yrityksessä käytetään. Vastauksien perusteella on havaittavissa, että yrityksiä käyttämät ratkaisut ovat usein sidoksissa toisiinsa. Esimerkiksi avoimien vastausten perusteella osa vastaajien yrityksistä käyttää pääasiassa tai pelkästään Microsoftin teknologioita ja osa taas Open Source teknologioita, joiden jakelu on vapaata ja käyttö pääsääntöisesti ilmaista. Pääsääntöisesti ilmaista siksi, koska panoksena näiden teknologioiden käytöstä kehittäjä jakaa oman kehitystyönsä, eli teknologian parissa käyttämänsä aikaa eteenpäin (Kaboli 2018).

Monivalintakysymykset toimivat lähinnä eri teknologioita ja metodeita arvottavina tekijöinä. Pääsääntöisesti työvälineen, eli esimerkiksi teknologisen ratkaisun tai menetelmän arvo muodostuu siitä, kuinka paljon sitä käytetään ja tarvitaan. Monivalintakysymysten vastaukset olivat huomattavasti avoimia vastauksia niukemmat, aivan kuin kyselyä suunnitellesani oletin. Niukkuus saattoi myös johtua siitä, että ennen monivalintakysymystä oli vastaaja jo paneutunut vastaamaan yhteen avoimeen kysymykseen mahdollisimman laajasti, eikä sitten enää kokenut oleelliseksi vastata seuraavaan kohtaan yhtä laajasti. Niukkuudesta huolimatta, näistä vastauksista sain paljon arvokasta tietoa asioista, jotka ovat tärkeitä, vaikka niitä ei välttämättä tule edes ajatelleeksi. Sen sijaan, että tuloksista muodostuisi yksinomaan massiivinen lista kuten avoimista vastauksista, monivalintakysymykset antoivat tietoa siitä, mikä on tällä hetkellä ohjelmistokehittäjälle tärkeää osaamista.

Tarkoituksenani on myös vertailla jonkin verran saamiani tuloksia aikaisemmin suoritettujen tutkimuksien tuloksiin selvittääkseni, onko tarpeilla selkeää muutosta jopa 20 vuoden takaisesta tarpeesta. Vertailu ei ole mahdollista yksi yhteen, koska tutkimukset eivät ole olleet identtiset. Ennemminkin vertailun tulokset voivat antaa viitteen muutoksista ja ehkä pienen mahdollisuuden ennakoida tulevaa, mihin en tässä kuitenkaan ota sen suuremmin kantaa.

5.1 Teknologiset ratkaisut

Työkaluilla tässä tapauksessa tarkoitetaan muun muassa tietokantoja, ohjelmointikieliä ja -kehyksiä sekä jossain määrin niitä sovelluksia, joilla itse varsinainen ohjelmointi tapahtuu. Eniten huomiota saivat aikaisemmissakin tutkimuksissa tärkeimmiksi osoittautuneet tietokannat sekä ohjelmistokehityskielet (kuvio 5). Yksikään monivalinnassa esitetty termi ei saanut kakkien 58 vastaajan tärkeysjärjestystä.



Kuvio 5: Yrityksen työkalut ja niiden tärkeys.

Suoraa vertailua aikaisemmin toteutettuihin tutkimuksiin on osin vaikea, ellei jopa mahdoton tehdä. Syynä siihen on se, että aikaisemmissa tutkimuksissa pääsääntöisesti on joko arvotettu kaikkia osaamisalueita massana (Ala-Mutka & Puhakka sekä Lethbridge) tai siten tulokset ovat mitanneet enemmän työkalun yleistä tarpeellisuutta (Teerimäki) yksittäisien kielten tai tekniikoiden sijaan. Viitteitä tuloksien samankaltaisuudesta antaa kuitenkin se, että moni eri ohjelmointikieli tuli usein valituksi tärkeäksi sekä tietokantojen tärkeys osoittautui suureksi.

Kyselyn avoimissa vastauksissa listattiin eri ohjelmointikieliä laidasta laitaan, mutta ehdottomasti eniten huomiota saivat 1995 alkunsa saanut Java (Habib 2016) eri ikäisine kehyyksineen, JavaScript kehyyksineen sekä C#. Finanssi- ja pankkialalla käytetty Cobol on edelleen tärkeää osaamista yhden kyselyyn vastanneen henkilön mukaan noin 60 vuoden iästään huolimatta (Cobol). Mobiiliohjelmoinnissa käytetyt nuorehkot kielet, 2011 julkaistu Android-kehityskieli Kotlin (Wellner 2017) sekä 2014 julkaistu iOS-kehityskieli Swift (Bohon 2017), saivat jonkin verran huomiota. Kotlinin suosio on noussut huomasti Googlen julkistettua tukensa Kotlinia kohtaan yhtenä virallisista Android-kehityksessä käytettävistä ohjelmistokehityskielistä vuonna 2017 (Protalinski 2017). Huomion määrä ei kuitenkaan vielä vahvista väitöstä siitä, että Kotlin olisi syrjäyttämässä Javan Android-mobiiliohjelmoinnissa (Laitila 2017). IOS kehityksessä käytettiin ennen Swiftin julkaisua pääsääntöisesti vain myös vastauksissakin muutaman kerran mainittua Objective-C kieltä, jota Swift on alkanut pikkuhiljaa syrjäyttämään (Austins 2017).

Tutkimuksen tulokset eivät antaneet varsinaista pohjaa ohjelmistokehittäjien välisissä yleisissä keskusteluissa tai tuloksissakaan esille tulleen väitteelle siitä, että jokin muu, kuten Scala, JavaScript tai Kotlin, korvaisi Javan. Yksi vastaaja oli kommentoinut työkaluja ja metodeita keräävään kohtaan kyselyssä ”Opetelkaa Scalaa” mikä on mielestäni oikein osuva esimerkki Java vs Scala keskusteluista. Javan ja Scalan viime vuosien kehitykseen liittyen Koskikallio kertoi aiheesta kysyessäni, uskoneensa aikaisemmin Scalan korvaavan Javan, mutta nykyään, Java 8 julkaisun jälkeen hän ei enää usko Scalan nousuun ainaakaan Javan korvaajana (Koskikallio, 2018). Väittämä JavaScript ohjelmistokehyyksien nopeasta kehityksestä (Koskikallio, 2016) saa jossain määrin jalansijaa, sillä JavaScript sekä sen monet ohjelmistokehyykset, kuten esimerkiksi 2011 keväällä julkaistu Angular (Gudelli 2017) ja alun perin 2013 julkaistu, 2016 valtavirtaan noussut React (Papp 2018), saivat paljon huomiota. Paikoin vastaajat mainitsivat vain kehyyksen jättäen itse JavaScriptin mainitsematta, mikä kertoo JavaScriptin yleistymisestä ohjelmistokehityskielenä erityisesti sen kehyyksien kehittymisen kautta.

Vaikka avoimeen kyselyn aloittaneeseen kysymykseen vain noin puolet vastaajista kirjasi jonkin tietokantaratkaisun, tutkimuksen kaikkien tuloksien yhteisanalyysin perusteella tietokannat osoittautuivat tärkeimmäksi yksittäiseksi työkaluksi, mitä odotinkin tuloksilta ainakin Ala-Mutkan sekä Teerimäen tutkimusten perusteella. Tietokantojen osaamistarpeet jakautuivat melko laajaan valikoimaan yksilöllisiä työkaluja, joista huomattavasti eniten huomiota saivat vuonna 1986 POSTGRES projektina alkanut, vuonna 1995 SQL-tulkin ja 1996 nykyisen nimensä saanut, PostgreSQL (Postresql) sekä 1995 julkaistu, kehittäjänsä Michael ”Monty” Wideniuksen My-tyttären mukaan nimetty MySQL (Widenius 2018). Jo vähemmälle käytölle jäänyt SolidDB sai myös joitain huomioita.

Tietokantojen tärkeys oli hajaantunut huomattavasti kapeammalle alalle yksittäisiä työkaluja, kuin ohjelmistokehityskielen kohdalla. Pääsääntöisesti tietokantaratkaisut ovat hyvin samankaltaisia, mutta hajonta kuitenkin kertoo erilaisien ja eri tavoin toimivien työkalujen tarpeellisuudesta. Tutkimuksen liitteenä on taulukko työkalujen tärkeyttä koskevan monivalintakysymyksen vastauksista, joka sisältää myös manuaalisesti lisätyt ”muu, mikä?” vastauksien työkalut (liite 2).

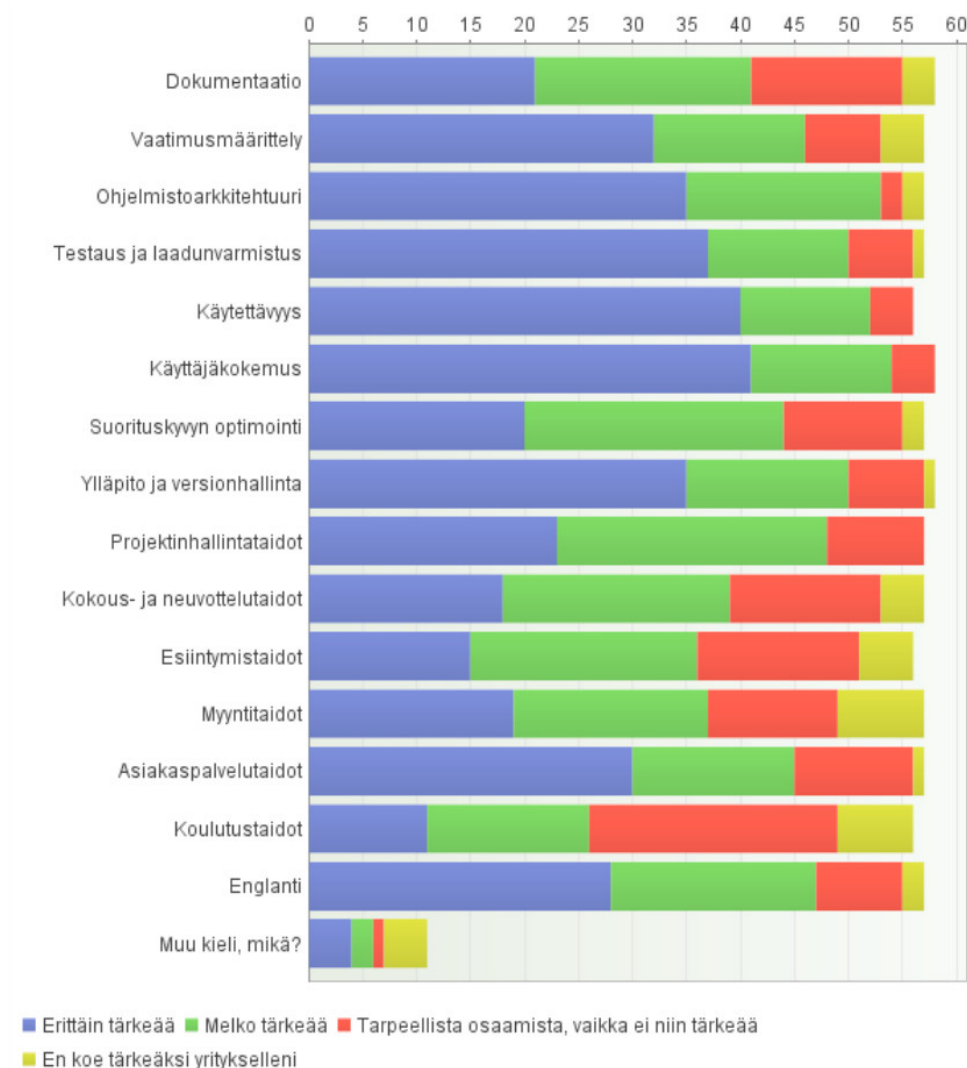
Noin viisi vuotta sitten, 2013 julkaistu open source sovellus Docker (Martin 2015) sai melkoisesti huomiota tässä tutkimuksessa, sillä jokainen sen kohdalla vastauksensa antanut vastaaja ilmoitti sen olevan vähintään jossain määrin tärkeä nyt tai ainakin ehkä tulevaisuudessa. Yksinkertaisimmillaan Docker on sovellus, jonka kautta voi saada minkä tahansa alustan sovelluksen toimimaan millä tahansa alustalla (Shabat 2016). Esimerkiksi Linux palvelimien pyörittämiseen ei välttämättä tarvitse olla enää Linux-ekspertti, sillä Docker hoitaa sen osuuden integroidulla käyttöliittymällään (Martin 2015).

Tutkimuksen tulokset antavat pohjaa ajatukselle siitä, että teknologioiden päivittymisestä huolimatta sen peruspiirteet pysyvät samana. Sovelluskehityksen peruspiirteinä voidaan mieltää sovelluksen mallintamisen ja suunnittelun, ohjelmoinnin sekä tietovaraston. Tietokannat ovat tärkeitä kaikissa tutkimuksissa, samoin ohjelmistokehityskielet – vaikkakin tässä tutkimuksessa tulokset ovat hajautuneet useamman eri kielen välille.

Huolimatta siitä, että agilet menetelmät olivat kyselyssä listattuina teknologiseen osaamiseen, ne ovat äärimmäisen tärkeä työväline projektinhallinnalle ja siten käsittelen niitä juuri projektihallinnallisesta näkökulmasta projektihallinnan yhteydessä seuraavassa aluvuossa.

5.2 Metodit ja toimintamallit

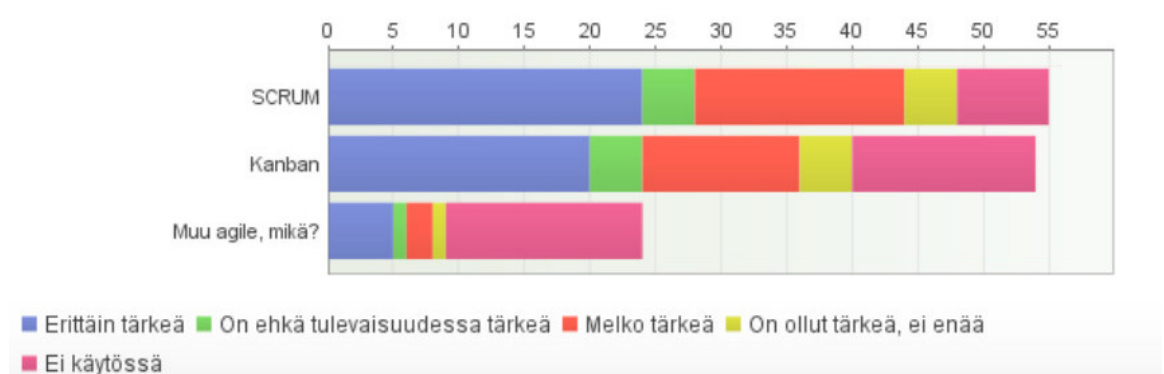
Metodit ja toimintamallit saivat vähemmän huomiota avoimissa vastauksissa kuin teknisemmät ratkaisut. Yleisesti huomiota saivat muun muassa versionhallinta, sosiaaliset taidot, esiintymiseen ja kommunikaatioon liittyvät taidot. Huomio jakaantui suhteellisen tasaisesti eri vaihtoehtojen välillä kuten alla olevasta kuviosta voi havaita (kuvio 6). Myös toimistotaidot saivat hieman huomiota, joten sen perusteella voimme olettaa niitäkin tarvittavan ainakin toisinaan ohjelmistokehitys-yrityksissä.



Kuvio 6: Yrityksen muu osaaminen ja sen tärkeys.

Muuhun osaamiseen liittyvissä asioissa eniten huomiota saivat käyttäjäystävällisyys, testaus ja laadunvarmistus sekä asiakaspalvelutaidot. Lähes kaikkia kyselyyn valittuja osaamis- ja taitoja pidettiin pääsääntöisesti tärkeinä tai melko tärkeinä, kuten tulostaulukosta (liite 5) selviää. Projektinhallintataitoja, käytettävyyttä ja käyttäjäkokemusta ei kukaan pitänyt yritykselleen tarpeettomana osaamisena. Tähän päätelmään kuitenkin jää lievä virhemarginaali, sillä yksi tai kaksi vastaajista ei antanut mielipidettään tarpeellisuudesta kaikkien kolmen osaamisalueen kohdalla.

Ohjelmistokehitystyön projektihallinnassa hyödynnetään usein agilea, eli ketterää kehitystä. Ketterässä kehityksessä olennaista on se, että projekti on jaettu pieniin iteraatioihin, joiden yksilöllinen kesto on 1-2 viikkoa (Kiiskinen, 2017) vaikka koko projekti kestäisi vuosia. Agileista menetelmistä esimerkkinä Scrum-kehityksessä iteraatiolla tarkoitetaan sprinttiä, jonka aikana kehittäjätiimi kokoontuu päivittäin muutaman minuutin palaveriin. Iteraation aikana sovitut työt on jaettu yksittäisiin pieniin tehtäviin, jotka löytyvät yleensä tiimin scrum-boardilta to-do, ongoing ja done statuksilla varustetuista kohdista esimerkiksi post-it lapuilla. Sprintin alussa on määritelty mitä kokonaisuuksia, joista tehtävät on pilkottu, sprintti sisältää ja lopussa käydään läpi sitä, miten sprintti meni (Poimala & Tolvanen, 2013). Osa vastaajista mainitsi vastauksessaan agilen tai jonkin sen alalajeista, kuten edellä esitellyn Scrumin, ja osa taas kertoi projektinohjauksen tapahtuvan pääsääntöisesti muun muassa palaverien tai keikkajärjestelmän kautta. Tämän pohjalta päätelen, ettei ole itsestäänselvyys, että projektit toteutetaan jollain opitulla tavalla, vaan jokainen yritys valitsee omaan toimintaansa tai tiettyyn projektiin soveltuvimman projektinohjausmenetelmän. Suurin osa vastaajista pitää Scrumia tai Kanbania tärkeänä työvälineenä, kuten alla olevasta työkalujen tärkeyttä mittaavasta kuviosta lohkaistusta vain agileen liittyvät vastaukset sisältävästä kuviosta (kuvio 7) voi päätellä.



Kuvio 7. Agilet menetelmät irrotettuna yrityksen työkaluja ja niiden tärkeyttä mittaavasta vastauksesta.

Käytännössä agilet menetelmät itsessään ovat työvälineitä, mutta niiden käyttö pohjautuu enemmän menetelmäosaamiseen. Ketterän kehityksen tärkein työväline on niin kutsuttu scrum-taulu, joka voidaan toteuttaa manuaalisesti fyysisenä tauluna post-it lappujen avulla tai virtuaalisesti esimerkiksi Trelloa kautta verkkotauluna. Trello on ketterän kehityksen työväline projektinhallinnassa, jota voi käyttää scrum-boardin tavoin internetin välityksellä. Trello, jota pidän itsekkin kätevänä työvälineenä projektien aikana saamani kokemuksen kautta, mainittiin 7 kertaa avoimen kysymyksen vastauksissa tärkeänä työvälineenä.

Moni vastaaja huomioi muun osaamisen yhteydessä olleessa avoimessa vastauksessaan tiimityön tärkeyttä eri tavoin. Tiimityö tai sitä vastaava sosiaalisuus oli mainittu kuusi kertaa kymmenestä tähän kohtaan jätetyssä vastauksessa. Yhden ihmisen ei oleteta osaa-
van ihan kaikkea, mutta tiimin voi odottaa osaavan yhdessä kaiken oleellisen. Sosiaaliset ja kommunikaationaaliset taidot koetaan tärkeinä ja ehkä osin myös osana yhteistyöky-
kyjä. Nämä vastaukset antavat viitettä siihen, että nykypäivänä panostetaan tiimin yhteis-
henkeen ja kokonaisosaamiseen huomattavasti enemmän kuin yksittäisiin ekspertteihin.

Tiimityön toimivuutta ajatellen on äärimmäisen tärkeää, että koko tiimi on perillä siitä, mitä juuri sillä hetkellä työstedään. Varsinkin projektin vetäjän näkökulmasta tiimin ollessa iso, voi olla haastavaa pitää kaikki kartalla eli tiimityön tukemiseen tarvitaan jo aikaisemmin tärkeäksi todettuja projektinhallinnallisia taitoja ja/tai esimerkiksi erilaisia ketterän kehityk-
sen menetelmiä. Isossa osassa yrityksiä projektit pyritään työstämään pääsääntöisesti aina ryhmä tai paritöinä. Yhteistyö ei yksinomaan helpota ja nopeuta tuloksen syntymistä toimivan tiimin kanssa, vaan se lisää myös vähemmän kokeneelle ohjelmistokehittäjälle mahdollisuuksia oppia toisilta mahdollisesti kokeneemmilta ohjelmistokehittäjiltä taitoja, joita hän työssään tarvitsee (Peurala 2018). Tästä voidaan päätellä, että tiimityö siis kul-
kee hyvin vahvassa yhteistyössä oppimisen ja itsensä kehittämisen kanssa.

Tiimityöllä on myös omat haasteensa etujen lisäksi. Mikäli tiimi muutoin osaa kaiken tii-
minä hyvin, mutta aikataulut eivät vain toimi, voi olla hyvin haasteellista sopia mahdolli-
sista tapaamis- tai kokousajoista niin, että kaikki olisivat kartalla siitä, mitä on milloinkin ta-
pahtumassa. Aikataulujen toimimattomuus voi esimerkiksi johtua eri työajoista. Etätyöpäi-
vät voivat myös luoda samankaltaista ongelmallisuutta, voi olla, että koodissa on jokin
pieni ongelma eikä etätyöpäivää pitävä ohjelmoija vaan osaa selittää sitä, kun toinen tii-
min jäsen saattaisi ratkaista ongelman jopa minuuteissa tuntien ihmettelyn sijaan. (Sipria
2018.)

Tiimityö kysyy ryhmältä tai parilta sosiaalisia taitoja. Usein tiimit muodostuvat henkilöiden kesken lähes automaattisesti, ihmisten välinen dynamiikka vaikuttaa parin tai tiimin muodostumisessa ja luonnollisesti muotoutuneet tiimit saavat yleensä parasta tulosta aikaiseksi (Toivanen 2018.) Osin tiimityön sivutuotteena olevat sosiaaliset taidot voidaan liittää myös myynti- ja asiakaspalvelutaitoihin. Kyselyn tulokset antavat viitettä sille, että ohjelmistokehittäjän toimiessa konsulttina, ovat asiakaspalvelutaidot usein tarpeellisia ja varsinkin pienien yritysten ollessa kyseessä voi tekijä päätyä myös myymään omaa ratkaisuun asiakkaalle joko itsenäisesti tai myyntitiimin jatkeena.

Ohjelmistoarkkitehtuuri, vaatimusmäärittely, testaus, ylläpito ja versionhallinta sekä design osoittautuivat kaikki melko tärkeiksi, sillä hieman yli puolet vastaajista oli sitä mieltä, että taidot ovat erittäin tärkeitä heidän yrityksissään. Kaikki edellä mainitut osaamisalueet ovat hyvin läheisissä tekemisissä toisiensa kanssa ja tuloksetkin antavat sille osviittaa. Näiden taitojen yhtäläinen tarpeellisuus saa viitettä myös tiimityön tarpeellisuudesta ja vastaajien kommentoinnista saa viitteitä siitä, että kaikkiin näihin yksilöllisiin taitoihin tarvittaisiin asiantuntija ja tiimissä näin myös todennäköisesti olisi.

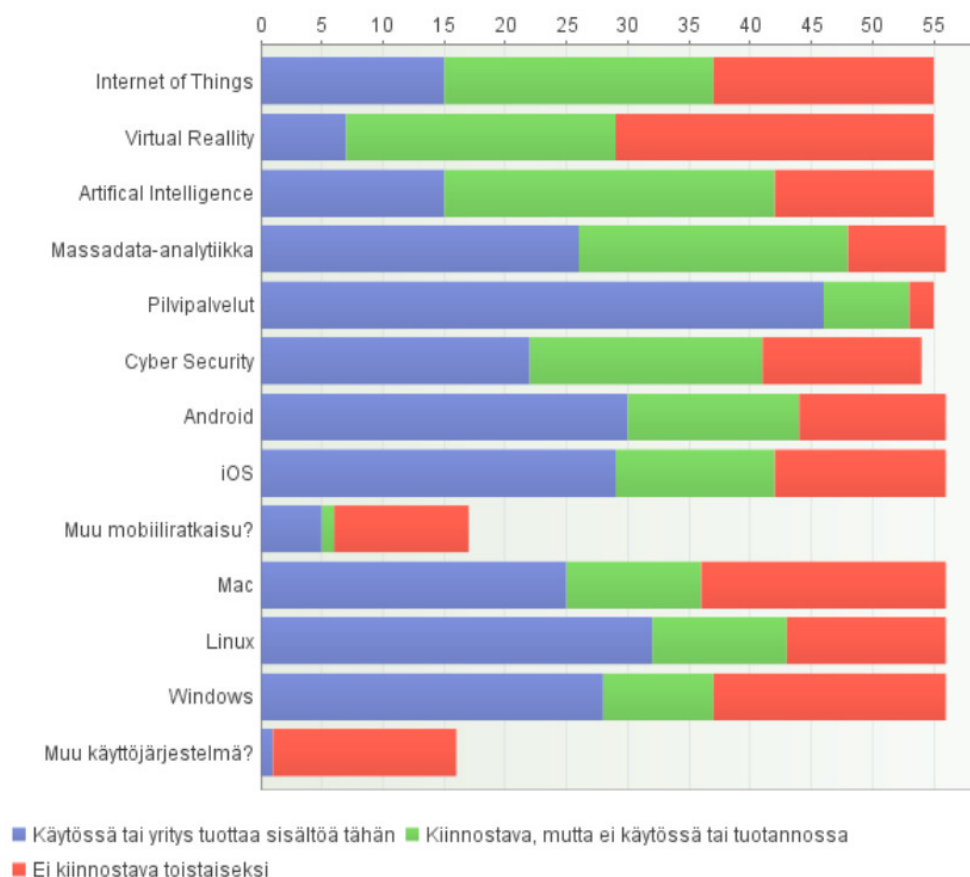
Versionhallinta, tai sen yleisin käyttöväylä Git, mainittiin tuloksissa yllätyksekseni vain muutamaan otteeseen. Käytännössä jokaisessa opintojeni aikaisessa projektissa on käytetty versionhallintaa. Versionhallinnan hyötyinä oman kokemukseni myötä voisin mainita esimerkiksi sen, että useampi eri kehittäjä voi päivittää kehitettävää sovellusta samanaikaisesti sekä eri versiot tehdystä sovelluksesta säilyvät tallessa, mikäli vanhaan versioon on syytä tai toisesta syytä palata. Näissä muutamissa vastauksissa kuitenkin painotettiin versionhallinnan tärkeyttä, joten sitä ei täysin voi sivuuttaa. Oman kokemukseni perusteella versionhallinta toimii usein sulavasti projektihallinnan, ketterän kehityksen, tiimityön sekä jopa dokumentaation kanssa käsi kädessä.

Dokumentaatiota pidetään myös oleellisena ja yksi vastaaja myöntääkin, että sitä on laiminlyöty tarpeesta huolimatta. Vastauksista löytyy viitteitä siihen, että dokumentaatiota, niin oleellista kuin se olisikin, laiminlyödään joko ajan tai rahan puutteen vuoksi. Ala-Mutkan ja Puhakan tutkimukseen nähden dokumentaation tarpeellisuudessa ei ole suurta eroa. Heidän tutkimuksensa tarpeellisuuslistauksessa dokumentaatio sijoittui kärkisijoille ja tähän tutkimukseen vastanneista noin kaksi kolmasosaa piti sitä tärkeänä tai melko tärkeänä.

Koska kysely oli toteutettu suomen kielellä, en kokenut oleelliseksi kielitaitotarpeita kysyessäni listata suomen kieltä kyselyyn. Silti, muutama vastaaja lisäsi suomen kielen vastauksiinsa, mikä kertoo sen tarpeellisuudesta, vaikka emme välttämättä ajattele suomen kielen olevan tarpeellinen taito. Englannin kieltä piti tärkeänä tai melko tärkeänä yli puolet vastaajista, vain kaksi ei pitänyt englannin kielen osaamista tärkeänä omalle yritykselleen.

5.3 Uudet teknologiat

Uudempiin teknologioihin paneudutaan hieman heikommin kuin jo tuttuihin vanhempiin teknologioihin, mikä on havaittavissa myös vastauksien määrässä, sillä yhteenkään vaihtoehtoon ei tullut 58 vastausta. Vain pilvipalvelut saavuttivat todellisen enemmistön kiinnostuksen kohteena 46 vastauksella. Mobiiliratkaisut sekä käyttöjärjestelmäratkaisut olivat tasaisesti tärkeitä noin puolelle vastaajista. Yleisesti otettuna uudempien teknologioiden käyttö ja kiinnostavuus jakautui muutoin aika tasaisesti, kuten seuraavasta kuviosta ilmenee (kuvio 8).



Kuvio 8: Yrityksen panostus uusiin teknologioihin.

Viime aikojen medioissa paljon huomiota saaneet IoT eli esineiden internet sekä AI eli keinoäly saivat vielä varsin vähän huomiota. Edistys on kuitenkin huimaa vielä noin kymmenen vuoden takaisesta Ala-Mutkan ja Puhakan tutkimuksesta. AI oli häntäpäässä käytettävyyden ja tarpeellisuuden suhteen, IoT ei itsessään edes päässyt mukaan listaukseen. Tosin, internet ja langattomat ohjelmistot, joita voisi jollain mittapuulla pitää IoT:n esivaatimuksena, olivat melko tarpeellisessa roolissa. Esivaatimus-ajatukseni juontaa siitä, että IoT on käytännössä laiteverkosto, johon kuuluu elektronisia laitteita, jotka keskustelevat keskenään ja välittävät dataa internetin välityksellä langattomasti yleensä pilvipalvelimeen, josta datan sisältöä päästään hyödyntämään (Rouse 2016).

Pilvipalveluiden suosio ja tarpeellisuus näkyy selkeästi tutkimuksen tuloksissa, sillä lähes jokainen vastaaja piti pilvipalveluita tärkeinä nyt tai ainakin tulevaisuudessa kiinnostavina. Vaikka terminä pilvipalvelu on ehkä vieras, moni käyttää pilvipalveluita joko päivittäin tai lähes päivittäin. Esimerkiksi pilvipalveluihin kuuluvaa Facebook:ia käyttää päivittäin noin 1,4 miljardia ihmistä (Mauno 2018). Muita pilvipalveluita ovat muun muassa useat sähköpostipalvelut, Instagram ja Skype (Kangasniemi & Lintulahti 2017.) Datan siirtyminen päätelaitteilta verkkoon voisi sanoa olevan entisestään yleistynyt trendi, sillä kuten IoT eivät pilvipalvelutkaan olleet aikaisemmin toteutetuissa tutkimuksissa mukana. Kasvua ei siis voi tulosten valossa arvioida, mutta viite kasvavaan tarpeeseen on olemassa, sillä lähes jokainen vastaaja piti pilvipalveluita tärkeinä nyt tai ainakin tulevaisuudessa kiinnostavina.

Erilaisien verkkopalvelujen määrän kehittyessä ja tietovarantojen siirtyessä internettiin kyberturvallisuudesta on hyvä huolehtia. Verkkoon siirtymisen trendiä ajatellessa kyberturvallisuudesta ollaan hyvin heikosti kiinnostuneita, sillä vain alle puolet vastaajista käyttää kyberturvallisuutta. Noin neljännes vastaajista ei pidä kyberturvallisuutta toistaiseksi kiinnostavana. Käytännössä kyberturvallisuutta yksinkertaisimmillaan voi toteuttaa olemalla hieman skeptinen; mikäli jokin vaikuttaa liian hyvältä ollakseen totta, se todennäköisesti ei ole totta, kuten sanonnassakin neuvotaan. Yleisin tapa joutua kyberturvallisuuden uhriksi on klikata jotain linkkiä tietämättä mitä tekee. (Forsman 2017.) Esimerkkinä vuoden 2017 lopulla Facebookissa kiersi PRISMA500 huijaus. Huijauksen tavoitteena oli saada henkilö vastaamaan kyselyyn, jonka päätteeksi hän huijauksen mukaan olisi saanut 500 euron lahjakortin. Todellisuudessa kyselyn täyttäjät päätyi kuukausimaksulliseen tilausansaamaan. Facebookissa seurasin itse tämän huijauksen leviämistä, moni tuttuni oli ikään kuin julkaissut PRISMA500 houkutteluviestin eri ryhmissä eikä välttämättä itse siis tiennyt linkin jakamisesta jouduttuaan ensin uhriksi klikkaamalla houkutteluviestiä aikaisemmin, vaikkei olisi kyselyyn vastannutkaan. (Kärkkäinen 2017.)

Todellisuudessa kyberturvallisuus ei siis ole täysin uusi asia. Huijauskirjeet ovat kiertäneet sähköposteissa ja jopa kirjepostissa ennen digitalisoitumisen aikakautta. Nykyään kun iso osa käyttämistämme palveluista on verkossa, pilvessä, kyberturvallisuuden uhat käyvät entistä vaarallisemmiksi. Huonolla tuurilla kyberrikoksen uhri päätyy tilausansa tai menettää esimerkiksi kalastelun kautta pankkitunnuksensa tai luottokorttinsa rikollisten käyttöön. IoT luo oman tarpeensa kyberturvallisuuden ymmärtämiseen, sillä jopa kodinkoneita voi valjastaa nykypäivänä palvelunestohyökkäyksiin (Forsman 2017.)

Myös monia listalta puuttuneita asioita listattiin lopuksi. Osa vastaajista jäi kaipaamaan vaihtoehtoihin muun muassa ohjelmistorobotiikkaa, Augmented Realityä eli lisättyä todellisuutta. Kokosin kaikki termit liitteenä olevaan taulukkoon (liite 6), joka sisältää uusia teknologioita koskevat vastaukset.

6 Pohdintaa

Aihe oli mielestäni aivan uskomattoman kiinnostava. Olisin voinut jatkaa sen läpikäyntiä ja sen aikana esiintyneisiin termeihin tutustumista läisyyskiin asti, ja todennäköisesti jatkan sitä vielä pitkään tämän opinnäytetyön valmistumisenkin jälkeen. Tulokset avarsivat valtavasti omaa maailmankatsomustani tulevaisuuden ohjelmistokehittäjänä, mikään ei ole vain mustaa ja valkoista. Se että joku sanoo asian olevan näin, ei tarkoita sen olevan kaikille niin, vaan aina on joku, joka on eri mieltä asioista. Tutkimukseen vastanneiden joukossa eri-ikäisten teknologioiden välinen suosio jakautui suhteellisen tasaisesti kumoten kuvitelmani siitä, että uudemmat teknologiat jyräisivät iäkkäämmät versionsa. Tähän pätee varmasti sanonta, toiset tykkäävät äidistä, toiset tyttärestä ja jotkut molemmista – aivan kuin kaikkiin muihinkin mielipide- ja kokemuseräisiin väittämiin.

Kun aloitin tutkimuksen suunnittelun 2016-2017 idean synnyttyä työharjoitteluni ensimmäisinä päivinä heinäkuussa 2016 en tiennyt puoliakaan tässä tutkimuksessa esiin tulleista termeistä. Tietotaitoni ja ymmärrykseni alaa kohtaan on kehittynyt valtavasti harppauksin varsinkin kuluneen vuoden aikana ohjelmistokehitys-alan työtehtävissä toimineiden ihmisten kanssa keskusteltuani, sekä tutkimuksen ja kaiken muun alaan liittyvän pyörittäessä ajatuksissani usein.

Kyselyssä ei ollut pakollisia kysymyksiä, vastaaja saattoi jättää vastaamatta vain ohittamalla kohdan. Tuloksia ei suoranaisesti voi verrata tai arvottaa tarkkoina tai varmoina tietoina, sillä jo vastaamatta jättäminen saattaa tarkoittaa sitä, ettei kyseinen asia ole käytössä tai sitä että se on niin itsestäänselvyys, ettei siihen vastaa sen tähden. Voi myös olla, että teknologia on itselle tärkeä ja oleellinen, muttei yrityksille, jolloin päätös termin merkityksellisyydestä on helpompi jättää tekemättä tai se voi olla puolueellinen omien intressejen mukaan. Kysely ei myöskään noudattanut klassisen tutkimuksen aineistonkeruutapaa, vastaajien valinta oli vain näennäisen sattumanvaraista. Houkuttelin vastaajia tutkimukselleni LinkedIn kanavallani, ja moni osoittikin mielenkiintoa kyselyä kohtaan. Kiinnostuneille jaoin linkin henkilökohtaisesti, eikä kyselyn rakennemallin vuoksi pysty mitenkään selvittämään kuka kyselyyn vastasi tai mitä väylää pitkin hän kyselyyn törmäsi. Rajasin ehkä hieman liiankin jyrkästi kyselyn jakelua, toisaalta rajausta oli hyvä - kaikki vastaajat olivat ohjelmistokehittäjiä tai toimivat ohjelmistokehitysyhtiössä, joten virheelliset tulokset vastaajan lähtökohtien vuoksi oli minimoitu. Talentedin verkoston yli tuhat potentiaalista vastaajaa toisaalta lisäävät satunaisuuden vaikutusta hieman. Tästä huolimatta

tulokset ovat lähinnä suuntaa-antavia, ehkä viitteellisen ennusteen tulevasta antavia tai osin trendejä kuvaavia, mutta eivät kuitenkaan täysin luotettavia mielenkiintoisuudestaan huolimatta.

Uskoisin, että ehkä yksi syy teknologioiden hajautumisessa on niiden jatkuva kehitys. Yhä useampi kehittäjä pääsee antamaan luovuudelleen siivet ja siten syntyy uusia kieliä sekä kehyksiä varsinkin open source kehitystyössä. Teknologian helpompi saatavuus edesauttaa yhtä lailla digitalisaation kasvua kuin teknologioiden kehittymistä, mikä aikaansaa eroja ja useampia erilaisia teknologisia ratkaisuja samasta aiheesta, kuten esimerkiksi JavaScriptin monet kehykset.

Ongelmallisena tutkimusta työstäessäni pidin kontaktoitumista. Aivan alussa, kun etsin vastaajia kyselyyni otin yhteyttä useisiin eri ohjelmistokehitys-yrityksien jossain johtoasemassa toimiviin henkilöihin ymmärsin kuinka ongelmalliseksi laajan vastaajamäärän haaliminen voisi muuttua. Kyselyä käynnistäessäni tavoittelin sataa vastausta kyselyyn, kuitenkin juteltuani Fraktion Peuralan kanssa ymmärsin osin, kuinka kunnianhimoinen tuo tavoite oli. Jos yritykselle tulee useasta lähteestä viikoittain jollei päivittäin pyyntöjä osallistua tutkimukseen, ei sen henkilökunnalla välttämättä ole aikaa tai intressejä vastata jokaiseen. Tästä syystä olenkin äärimmäisen iloinen 58 kattavasta vastauksesta, jotka sain kahden kuukauden aikana. Korkeat tavoitteet luovat korkean kynnyksen.

6.1 Havainnot

Tuloksia tulkitessani tein mielestäni äärimmäisen mielenkiintoisia havaintoja. Havaintoni perustan osin omaan kokemukseeni, tämän ja vanhempien tutkimuksien tuloksiin sekä kirjoitettuun teoriaan aiheesta.

Yrityksien koko ja ikäeroilla on havaittavissa lievää yhteyttä. Kaikkiaan suurin osa yrityksistä oli 51-250 henkilöä työllistäviä ja ainoastaan yli kymmenvuotiaiden yritysten joukosta löytyi henkilöstömäärältään tätä suurempia yrityksiä. Näiden tulosten valossa siis voisin väittää, että pääasiassa nuoret yritykset ovat pienempiä ja suurin osa isoista yrityksistä on toiminut alalla kauemmin. Kuitenkin, kuten aikaisemmin mainitsin, tämän tutkimuksen tulokset ovat lähinnä suuntaa-antavia, joten tälle olettamalleni ei ole kuitenkaan vahvaa perustaa. Lisäksi vastaajia oli alle 60, mikä on mikro-osa tämän maan ohjelmistokehittäjistä ja yrityksistä, joissa he työskentelevät.

Pidän ehkä tärkeimpänä yleishavaintona sitä, että se mikä on kaikkein tärkeintä meille, sitä emme ajattele tai mieti. Tämä väitteeni pohjaa oikeastaan kaikkeen; sekä ohjelmistokehittäjän työelämän taitotarpeisiin, että päivittäiseen elämäämme. Osa tarvittavista asioista on yrityksille jopa niin tärkeää, ettei niiden olemassaoloa varsinaisesti tarvitse ajatella tai muistaa, koska niitä käytetään kuitenkin jatkuvasti. Näitä tarpeita voisi kuvata kirjoitustaitona, joka on ehdottoman tärkeä, mutta kuinka moni todellisuudessa ajattelee kirjoittamista taitona, joka on tärkeä hänen työpaikallaan. Loistavana esimerkkinä tärkeimpien asioiden muuttumisesta itsestäänselvyyksiksi voin esittää sen, että vain yksi kyselyyn vastanneista listasi avoimessa vastauksessaan myös tietokoneen, jota voimme pitää automaattisena osana työtä, sillä se on ohjelmistokehittäjän tärkein työkalu, mutta sitä ei kuitenkaan ajatella varsinaisena työkaluna.

Laajan avoimen kysymyksen ja tärkeyttä arvottavien monivalintakysymyksien välisestä analysoinnista havaitsin moniakin mielenkiintoisia seikkoja. Tärkeimpänä näistä tietokannat, jotka usein ovat koko sovelluksen ja sen ympärillä käyvän kehityksen kivijalka, unohdettiin avoimessa kysymyksessä lähes täysin. Tietokannat tai jonkin sen käyttökehyksen oli maininnut vain hieman yli puolet kaikista vastaajista, kun taas tärkeyttä arvottavassa monivalintakysymyksessä vain yksi vastaaja ilmoitti, etteivät tietokannat olisi käytössä ollenkaan. Yksikään tähän kohtaan tärkeyttä arvottavaan kysymykseen vastanneesta 49 vastaajasta ei ollut sitä mieltä, etteivätkö ne olisi enää tärkeitä tai että ne saattaisivat ehkä olla tulevaisuudessa tärkeitä. Tietokantoja piti tällä hetkellä oman työnsä kannalta tärkeänä 98% vastaajista.

Tietokannat, joita siis pidetään äärimmäisen tärkeänä, saatetaan täysin unohtaa, kun miettii mitä osaamista oikeasti tarvitaan. Oman kokemuksenikin mukaan suurin osa kaikista sovelluskehityksestä käyttää jonkinlaista tietokantaratkaisua. Voisinkin kysyä tähän liittyen mitä hyötyä olisi vaikka verkkokauppasovelluksesta, joka ei talleta tilauksia tai myynnissä olevia tuotteita mihinkään. Vaikka kyseisen kaltainen sovellus toimisikin rajapinnan kautta ilman omaa tietokantaa, se silti hakee tiedon jostain tietokannasta ja lähettää tietoa johonkin tietokantaan.

Toinen jopa silmiinpistävä havainto liittyy erityisesti ohjelmistokehityskieli JavaScriptiin. JavaScript sai kaikista eniten mainintoja avoimessa kysymyksessä, mutta osa vastaajista saattoi mainita vain yhden tai useamman JavaScriptin kehyksen kuten Angularin tai JQuery:n, mainitsematta itse ohjelmistokehityskieltä. Tulosten perusteella JavaScriptin suosio siis johtuu sen kehyksien kehityksestä. JavaScript kehityksestä puhuttaessa usein

puhutaankin siitä, millä kehyksellä sitä käytetään. Esimerkiksi mobiiliohjelmoinnin kurssilla en puhunut JavaScript ohjelman ohjelmoinnista vaan kerroin tekeväni sovellusta JQueryllä, eli sorrun itsekin tähän tärkeän pohjan unohtamiseen herkästi.

Versionhallinnan olemassaoloa ei pääsääntöisesti edes mainittu tärkeydestään huolimatta. Jokainen sovellus, jonka toteuttamisessa olen ollut mukana, on jossain määrin käyttänyt versionhallintaa. Versionhallintaa sen alkukantaisimmassa muodossa on jopa se, että kehitettävästä sovelluksesta on useampi eri tallennusversio vaikkei sitä yleisesti pidetäkään virallisesti versionhallintana, mutta yleisimmin jopa yksilökoodaamisessa käytetään paljon versionhallinnan työkalua kuten Git:iä. Versionhallinta ei pelkästään säilytä vanhempia versioita sitä varten, jos niihin tarvitsee palata, vaan se myös helpottaa suunnattomasti sovelluksen kehittämisen eri henkilöiden toimesta samaan aikaan. Tätä ei kuitenkaan ajatella kokoaikaisesti. Versionhallinta saattaa olla jopa niin rutiininomaista, ettei ohjelmistokehittäjän edes tarvitse miettiä sen toimintaa vaan lähettää tuotoksensa automaattisesti esimerkiksi Git:iin.

Sosiaaliin taitoihin liittyvän huomion muassa olettamukseni siitä, että perinteisempi introverttihenkinen sovelluskehitys olisi jäämässä historiaan saa pohjaa. Tiimityöskentelyn tarpeellisuutta painotetaan paljon, yrityksissä toteutetaan pääsääntöisesti pari tai ryhmätyötä ja harvemmin kenenkään annetaan jäädä pitkäksi aikaa hakkaamaan yksin päätään seinään ongelman edessä. Kehittäjiä kannustetaan luomaan sosiaalisia kontakteja omalla työpaikallaan sekä myös verkostoitumaan muuallakin. Tieto kulkee suusta suuhun ja uudet asiat saavat kiinnostusta juuri tätä kautta, jos jättäytyy pitkäksi aikaa kontaktien ulottumattomiin saattaa yhtäkkiä huomata jääneensä jälkeen nopeasti kehittyvän teknologian aallonharjalta.

Vuosien saatossa tietotekniset ratkaisut kehittyvät jatkuvasti, mikä tuntuukin olevan alan vahvin teema – mikä uskoakseni tulee tässäkin tutkimuksessa vahvasti ilmi. Vajaan kahdenkymmenen vuoden aikana kehitys on ollut jopa huimaa, jotkin asiat ovat säilyttäneet oman tärkeytensä ja toiset ovat nousseet uusina ratkaisuin vanhojen tilalle. Esimerkiksi dokumentaation tarve vaikuttaa olevan määrällisesti nousemassa, sillä Lethbridgen 20 vuoden takaisesta tutkimuksesta en havainnut selkeää tarvemääritystä dokumentaatiolle, mikä tänä päivänä kuitenkin koetaan tarpeellisenä, kuten seuraavasta taulukosta, joka sisältää luvussa 3 esittelemäni taulukon sisältäen myös tämän tutkimuksen tulokset (taulukko 3). Dokumentaation tarpeellisuuden kasvun oletan syntyvän entistä

nopeatempoisemmasta kehityksestä sekä siitä, että useampi ihminen käyttää erilaisia teknologisia ratkaisuja ja siten tarvitsee enemmän dokumentaatiota niiden käyttöön ja jatkekehitykseen.

	Lethbridge, 1999	Ala-Mutka & Puhakka, 2009	Teerimäki, 2012	Kalliosto, 2018
Ohjelmoinnin perustaidot	-	1	A 2	-
Ohjelmistokehityskieli	1	2	N / A	N / A
Tietorakenne / tietoverkot	2	10	A 8	-
Määrittely ja suunnittelu	3	24	A 10	B 6
Sovellusarkkitehtuuri	4	20	A 11	B 5
Vaatimusmäärittely	5	5	A 7	B 6
Käytettävyys	6	12	-	B 2
Esiintymistaidot	10	9	B 3	B 14
Projektinhallinta	11	11	A 4	B 9
Testaus ja/tai laadunvarmistus	12	8	A 6	B 3
Käyttöjärjestelmät	15	17	A 5 **	N / A
Tietokannat	16	3	A 1	A 1
Dokumentointi	21 *	4	B 1	B 10
Kokous ja neuvottelutaidot	25	9	B 4	B 13
Tietoturva	42	21	A 9	C 8
Internetpalvelut, www-sovellukset	-	16	A 3	C 1 ***

* File Management, ** Käyttöliittymät, *** Pilvipalvelut

Taulukko 3. Kaikkien tutkimuksien välistä vertailua. Tutkimusten eroavaisuuksien vuoksi tämän tutkimuksen tulokset on jaettava 3 eri kategoriaan, jossa internetpalvelut korvaavat tämän tutkimuksen suosituimmalla internetiä hyödyntävällä osaamisalueella, Pilvipalveluilla.

Taulukon eriäviä tietoja voi selittää myös se, miten ja mitä muuta on kysytty. Lethbridgen sekä Ala-Mutkan ja Puhakan tutkimuksissa kysyttiin kaikki asiat erikseen jaottelemana kerralla. Teerimäen tutkimuksessa taas aiheet oli jaoteltu kahteen eri osa-alueeseen, jotka taulukossani jaoin A ja B ryhmään. Omassa tutkimuksessani olin edeltäjäni enemmän perehtynyt yksittäisiin teknologioihin, jonka johdosta jaottelu oli Teerimäen tutkimukseen verrattuna hyvinkin erilainen, lisäksi oman tutkimukseni tuloksien taulukointiin tarvitsin kolmannen ryhmän C:n eli uudet teknologiat.

Toinen mielenkiintoinen havainto tutkimuksien välillä on tietokantojen tarpeellisuuden kehitys, vuonna 1999 lähes 20 vuotta sitten Lethbridgen tutkimuksen mukaan tietokannat olivat 16. tärkein aihe, noin 10 vuotta myöhemmin Ala-Mutkan ja Puhakan tutkimuksessa kolmanneksi tärkein ja vain kolmea vuotta myöhemmin tietokantojen tärkeys oli kirinyt ensimmäiselle sijalle Teerimäen tutkimuksessa.

Internetpalveluista, joita voi pitää oman tutkimukseni mukaan melko tärkeinä pidettyjen IoT:in ja pilvipalveluiden esimuotona, tai kyberturvallisuudesta, jota pidetään nykyään kohtalaisen tärkeänä, ei ollut juuri edes tietoa noin 20 vuotta sitten. Kymmenen vuotta

myöhemmin 72 aihetta arvottavassa Ala-Mutkan ja Puhakan tutkimuksessa internetpalvelut nousivat 25 tärkeimmän aiheen joukkoon ja Teerimäen tutkimuksessa ne olivat jo 10 tärkeimmän aiheen joukossa. Lisäksi mielenkiintoisena havaintona Lethbridgen tutkimuksessa robotiikka oli 75 aiheesta toiseksi vähiten tärkeänä koettu, tänä päivänä robotiikka selkeästi kiinnostaa enemmän, vaikka se ei saanutkaan kuin muutaman maininnan. Mikäli olisin maininnut vaihtoehtoisesti robotiikan uskon, että se olisi saanut huomattavasti enemmän huomiota. Näiden havaintojen pohjalta voin väittää, että internetpalvelut sekä tietoturva tulevat olemaan tulevaisuudessaakin tärkeitä ja ehkä edes kahtiajako teknologisista muuhun osaamiseen ei enää ole kohta riittävä vaan jako olisi syytä toteuttaa teknologian, muun osaamisen ja verkkopalveluiden sekä robotiikan, johon keinoälykin osin kuuluu, välillä.

6.2 Mahdollisuuksia

Yhtenä lähteenäni toiminut Teerimäki päätti opinnäytetyönsä loppusanat mielipiteeseen, jonka mukaan tutkimus olisi syytä toistaa muutamien vuosien välein, jotta tieto pysyisi ajantasaisena. Tutkimustani suunnitellessani en tiennyt vielä Teerimäen tutkimuksesta, mutta tämä mielipide yhdistää molempia tutkimuksia täysin. Olin samaa mieltä hänen kanssaan, tietämättäni hänen tutkimuksestaan, jo ennen kuin lähdin edes todella toteuttamaan tätä tutkimusta. Mielestäni olisi vain kaikkien edun mukaista, jos vastaavan kaltainen tutkimus suoritettaisiin aina muutaman vuoden välein, koska silloin tieto tarpeista olisi jatkuvasti ajantasaista.

Voisin myös kuvitella tänä digitalisaation aikakautena, että tämänkaltaisen kyselyn voisi toteuttaa aina muutaman vuoden välein, siten että yrityksiltä kysyttäisiin heidän käyttämiään teknologioita. Ehkä juuri tähän tiedonkeruutarkoitukseen voisi jopa kehittää jollain kurssilla sovelluksen, niin ettei tiedon keräämiseen ja sen läpi kahlaamiseen tarvitsisi käyttää ylenpalttisesti aikaa, muun muassa uuden kyselyn luomisen tai haastatteluiden vuoksi. Kerätyn datan mukaan voitaisiin kehittää tietoa siitä, mihin opiskelijoita kannattaisi kannustaa perehtymään. Esimerkiksi opinto-oppaaseen voisi päivittää johdannon kaltaisen sivun, joka sisältää tietoa käytetyistä teknologioista, ilman opintokokonaisuuksien kuvaamiseen puuttumista. Sillä tavoin tieto mahdollisista osaamistarpeista olisi helposti opiskelijan saatavissa.

Teknologioita on valtavasti ja niiden käyttö riippuu paljon makuasioista. Luonnollisesti vaihtuvuutta on paljon mutta osa teknologioista jää kuitenkin elämään pitkäksikin aikaa.

Nopean kehityksen mukana ei ole helppo olla, mutta osin sille ehkä olisi tarvetta ja jopa kysyntää. Voisinkin kuvitella, että etuna opiskelijoille olisi muutama vapaavalintainen tai vaihtoehtoinen kurssi, joiden sisältö rakentuu kurssin alkaessa. Sen sijaan, että olisi määriteltä tietyn kielen kanssa toimimista voisi ryhmä yhdessä ehdottaa mitä he haluaisivat oppia. Tässä yhteydessä opiskelijoille voisikin tarjota vastaavanlaisesta kyselystä kootun teknologiaratkaisujen listan, ja he voisivat perehtyä eri teknologisiin ratkaisuihin ja sitten ehdottaa omaa suosikkiaan kurssilla opiskeltavaksi. Näin lehtorikin voisi ottaa tilaisuudesta vaarin ja tutustua itselleenkin uuteen teknologiaan opiskelijoiden kanssa, sillä uskon jo kokeneemman lehtorin ymmärtävän uuden teknologian vähemmällä vaivalla kuin uransa alulla oleva opiskelija voi ymmärtää, ja siten pystyvän neuvomaan opiskelijoita, miten tuota teknologiaa ehkä voisi hyödyntää.

Tämän tai tämänkaltaisen tutkimuksen tuottamaa listaa voi myös käyttää aivan opintojaan aloittavien peruskurssien materiaalina tai esimerkiksi opinto-oppaan liitteenä. Oman opintopolkuni Haaga-Heliassa aloitti tietotekniset välineet -kurssi, joka toimi jossain määrin opintojen esittelykurssina. Vastaavanlaisella esittely- tai tutustumiskurssilla voi mahdollisesti avata eri teknologioita hieman opiskelijoille, mikä auttaisi opiskelijaa ehkä etsimään tietoa lisää itseään kiinnostavista osaamisalueista. Omakohtaisesti, jos olisin tietänyt opintojani aloittaessa kaiken, mitä nyt tiedän, olisin jo silloin suuntautunut enemmän projekti-johtamiseen ja keskittynyt enemmän tietokantojen kanssa toimimiseen. Lisäksi olisin varmasti ollut enemmän kiinnostunut JavaScriptistä jo aikaisemmin ja siten ehkä oppinutkin sitä hieman paremmin.

Opintojen aikana konseptien ja käsitteiden parempi oppiminen voi edesauttaa enemmän uusien teknologioiden oppimisessa ja haltuunotossa. Tämän vuoksi ehkä kannattaisi painottaa vielä enemmän juuri konsepteja sekä käsitteitä eri teknologioiden sijaan, mutta tarjota tieto olemassa olevista teknologioista jo opintojen alkupuolella.

6.3 Oma oppimiskokemukseni

Tutkimuksen aloittamisen syy sekä itse tutkimus olivat mielestäni äärimmäisen mielenkiintoisia, tästä johtuen käytn varmasti sen työstämiseen aikaa jopa enemmän kuin se olisi vaatinut. En kuitenkaan kadu käyttämäni aikaa missään tapauksessa, erinäisiin pikku-seikkoihin huomion kiinnittäminen loi minulle valtavasti uutta tietoa ja ymmärrystä. Uuden oppiminen on aina ollut intohimoni, tällä alalla intohimoni kohtaa urani mahdollisuudet ja tutkimuksen toteuttaminen on vain kasvattanut tiedonnälkääni ja vahvistanut

ymmärrystäni siitä, mitä oikeasti haluan tehdä. Olen koko opintojeni ajan ollut äärimmäisen kiinnostunut tietokantojen toiminnallisuudesta ja käyttömahdollisuuksista, ja tämän tutkimuksen tehtyäni ymmärrän nyt osin jopa miksi. Nautin tiedon keruusta ja käytöstä sekä sen analysoinnista. Ehkäpä tämän tutkimuksen toteuttaminen oli herättäjäni ja lähdän vielä jatkamaan opintojani data-analyttikoksi tai tutkijaksi asti.

Tutkimuksen idean synnyttyä se vain kasvoi ja kasvoi, kunnes lopulta päätin tehdä siitä opinnäytetyöni. En kadu päätöstä aiheen työläydestä huolimatta.

Tutkimus oli laaja, mikä aiheutti välillä jopa harmaita hiuksia. Olisin voinut rajata tutkittavat asiat niiden päämuotoihin, kuten eri kielten sijasta ohjelmistokehityksen kieliin yleisesti, mutta silloin taas en olisi saanut näin valtavasti mielekästä tietoa käyttööni. Tuloksien keruu oli osin haasteellista, minkä ei pitäisi olla yllätys. Monikaan meistä ei ole innostunut käyttämään aikaa johonkin, josta emme usko hyötyvämmme itse, ainakaan lähitulevaisuudessa. Näin laajan tutkimuksen olisi voinut hyvinkin toteuttaa tutkimuskumppanin kanssa, mutta silloin en olisi voinut keskittyä asioihin juuri siten kuin tutkimuksen teon aikana pääsin keskittymään. En olisi välttämättä hakenut tietoa esimerkiksi yllättävänkin iäkkään PostgreSQL:n julkaisusta, vaan saattaisin edelleen kuvitella sen olevan täysin 2010-luvun tuotos.

En usko, että tekisin juurikaan asioita toisin, jos lähtisin nyt toteuttamaan tutkimusta uudelleen. Se mitä muuttaisin liittyisi todennäköisesti eniten kyselyyn, jolla keräsin tietoa tutkimusta varten. Saattaisin muuttaa hieman järjestystä termeissä, niiden määrää ja laatua, eli kysyisin ehkä hieman yksityiskohtaisempia kysymyksiä, silläkin uhalla, että kyselyyn vastaamiseen kuluisi enemmän aikaa ja vastaajia olisi vähemmän. Tuloksia tutkiessani huomasin, että muun muassa SCRUM aiheiset kysymykset olisi pitänyt siirtää työkaluista muuhun osaamiseen ennen kyselyn julkaisua, sillä agile toimii useimmiten yhteistyössä projektihallinnan - joka on muussa osaamisessa - kanssa.

Omaa toimintani kehittääkseni tärkein kehityskohde olisi työskentelyn yhtäjaksoisuus. Sen sijaan että toteutin tutkimuksen läpikäyntiä sykleissä, olisin ehkä päässyt helpommalla sen jokapäiväisessä työstössä sen aktiivisimman ajan läpi. Sykleissä toimiminen kuitenkin auttoi motivaation kehitystä läpi tutkimuksen toteutuksen. Pääsääntöisesti tätä tutkimusta oli joka tapauksessa aivan mahtava päästä toteuttamaan.

Lähteet

Ala-Mutka, K. & Puhakka, A. 2009. Survey on the Knowledge and Education Needs of Software Professionals. Tampereen teknillinen yliopisto, ohjelmistotekniikan laitos. Luettavissa: http://www.cs.tut.fi/ohj/laitosraportit/laitosraportti_alamutka_puhakka.pdf. Luettu: 23.4.2018.

Ali-Mattila, V. 23.11.2016. Lakkaa polttamasta yrityksen rahoja useita tuhansia euroja! VM IT OY blogi. Luettavissa: <http://www.vmit.fi/blogi/2015/11/23/lakkaa-polttamasta-yrityksen-rahoja-useita-tuhansia-euroja>. Luettu 14.10.2016.

Austins, C. 4.4.2017. Opinion. Top 6 Programming Languages for Mobile App Development. Luettavissa: <https://dzone.com/articles/top-6-programming-languages-for-mobile-app-develop>. Luettu: 25.4.2018.

Bohon, C. 7.6.2017. Apple's Swift programming language: Cheat sheet. Luettavissa. <https://www.techrepublic.com/article/apples-swift-programming-language-the-smart-persons-guide/>. Luettu: 25.4.2018.

Brown, P. 25.1.2017. 20 Billion Connected Internet of Things Devices in 2017, IHS Markit Says. Luettavissa: <http://electronics360.globalspec.com/article/8032/20-billion-connected-internet-of-things-devices-in-2017-ih-markit-says>. Luettu: 25.4.2018.

Cobol. The National Museum of American history. Luettavissa: <http://americanhistory.si.edu/cobol/introduction> Luettu: 25.4.2018.

Forsman, T. 9.4.2017. Kyberturvallisuus on jo kaikkien ongelma – "Joissain maissa ihmiset käyvät töissä tehtailemassa huijauksia". Osin maksullisen sivun artikkeli. Luettavissa: <https://www.ksml.fi/kotimaa/Kyberturvallisuus-on-jo-kaikkien-ongelma-%E2%80%93-Joissain-maissa-ihmiset-k%C3%A4yv%C3%A4t-t%C3%B6iss%C3%A4-tehtailemassa-huijauksia/963894>. Luettu: 13.5.2017.

Gudelli, A. 26.3.2017. History of AngularJs. Luettavissa: <http://www.angularjswiki.com/angularjs/history-of-angularjs/>. Luettu 2.5.2018.

Habib, O. 20.12.2016. The History and Future of Java Programming Language. Appdynamics blogi. Luettavissa: <https://blog.appdynamics.com/engineering/the-history-and-future-of-java-programming-language/>. Luettu: 25.4.2018.

Hakkarainen, P. 24.4.2018. Product Development Manager. Mediconsult Oy. Haastattelu, Vantaa.

Jäntti, J. 2012. Suomalaisen pk-yrityksen tietotekniikka-asiantuntijan työ- ja osaamisprofiili. Amk-opinnäytetyö. Haaga-Helia ammattikorkeakoulu. Helsinki. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/47001/Opinnaytetyo_Jantti_Jouni.pdf;jsessionid=69DD275BAF6CCCE39BAE80C70D014C3B?sequence=1. Luettu 14.10.2016.

Kaboli, S. 4.4.2018. Lead Software Engineer & Solution Architect, Tieto Oyj. Luento, Tieto Boost for Techies Espoo, Tieto Keilalahti office.

Kangasniemi, H & Lintulahti, M. 10.1.2017. Mikä on pilvipalvelu? Luettavissa: <https://yksityisille.hub.elisa.fi/mika-on-pilvipalvelu/>. Luettu 14.5.2018.

Karenko, J. 6.10.2016. Ohjelmistokehittäjä. Fn(x) Solutions Oy. Haastattelu. Helsinki.

Kiiskinen, J. 14.2.2007. Moderni ohjelmistokehitys - vesiputousmalli vs. ketterät menetelmät. Luettavissa: <http://blogi.sysart.fi/moderni-ohjelmistokehitys-pahkinankuoressa-vesiputousmalli-vs.-ketterat-menetelmat>. Luettu: 1.5.2018.

Kinnersley, B. 8.1.2018. The Language List. Arkisto. Luettavissa: <https://web.archive.org/web/20160506170543/http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>. Luettu: 24.4.2018

Koskikallio, E. 10.10.2016. Ohjelmistoarkkitehti. Fn(x) Solutions Oy / Null Pointer Oy. Haastattelu. Helsinki.

Koskikallio, E. 12.4.2018. Ohjelmistoarkkitehti. Null Pointer Oy. Haastattelu. Helsinki.

Kärkkäinen, H. 24.10.2017. Varo Prismän nimissä kiertävää viestiä – huijarit yrittävät kaapata Facebook-tilisi. Luettavissa: <https://www.is.fi/digitoday/tietoturva/art-2000005421198.html>. Luettu: 13.5.2017

Laitila, T. 12.10.2017. Hyvästi Android-java – kotlin jyrää. Luettavissa: https://www.tivi.fi/Kaikki_uutiset/hyvasti-android-java-kotlin-jyraa-6681995. Luettu 25.4.2018.

Lethbridge, T. Heinäkuu 1999. The Relevance of Education to Software Practitioners: Data from the 1998 Survey. University of Ottawa Computer Science Technical Report TR-99-06. Luettavissa: <http://www.site.uottawa.ca/~tcl/edrel/EdrelTechReport.pdf>. Luettu: 14.5.2018.

Lévéné, É. 22.10.2017. Languages. Kuva. Luettavissa: <https://www.leve-nez.com/lang/lang.pdf/>. Luettu: 15.5.2018.

Martin, N. Toukokuu 2015. A brief history of Docker Containers' overnight success. Luettavissa: <https://searchservervirtualization.techtarget.com/feature/A-brief-history-of-Docker-Containers-overnight-success>. Luettu: 11.5.2018.

Mauno, H. 31.1.2018. Facebookin käyttäjämäärä kasvoi laimeasti, tulos yli odotusten. Luettavissa: <https://www.kauppalehti.fi/uutiset/uutinen/nyjuyawY>. Luettu: 14.5.2018.

Papp, A. Maaliskuu 2018. History of React.js on a Timeline, Blogi. Luettavissa: <https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>. Luettu: 2.5.2018.

Peurala, J. 14.5.2018. Co-Founder, CEO. Fraktio Oy. Haastattelu, Vantaa.

Poimala, S & Tolvanen, P. 6.6.2013. Ketteryys haltuun: Scrum pähkinäkuoressa. Luettavissa: <https://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-scrum-pahkinankuoressa/>. Luettu 1.5.2018.

Postgresql. A Brief History of PostgreSQL. Luettavissa: <https://www.postgresql.org/docs/8.4/static/history.html>. Luettu: 1.5.2018

Protalinski, E. 5.2.2018. Google launches Android KTX preview, a set of Kotlin extensions for developers. Luettavissa: <https://venturebeat.com/2018/02/05/google-launches-android-ktx-preview-a-set-of-kotlin-extensions-for-developers/>. Luettu: 25.4.2018.

Recommendation of the European Parliament and of the council on key competences for lifelong learning. Luettavissa: <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:c11090>. Luettu: 14.10.2016.

Rouse, M. Huhtikuu 2016. Internet of Things (IoT). Luettavissa: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT/>. Luettu: 11.5.2018

Ryan, R & Deci, E. 2000. Intrinsic and Extrinsic Motivations: Classic Definitions and New Direction. Contemporary Educational Psychology 25, 54–67. University of Rochester. Luettavissa: http://selfdeterminationtheory.org/SDT/documents/2000_RyanDeci_IntExtDefs.pdf. Luettu: 16.9.2016.

Shabat, A. 19.10.2016. Docker in a nutshell - Part 1: What is Docker? LinkedIn artikkeli. Luettavissa: <https://www.linkedin.com/pulse/docker-nutshell-part-1-what-asaf-shabat/>. Luettu 11.5.2018.

Sipria, M. 24.4.2018. Senior Software Developer. Solinor and Payment Highway. Haastattelu, Vantaa.

Soderbery, R. 7.1.2013. Cisco Executive. How Many Things Are Currently Connected To The "Internet of Things" (IoT)? Luettavissa: <https://www.forbes.com/sites/quora/2013/01/07/how-many-things-are-currently-connected-to-the-internet-of-things-iot/#725dea2ebd2d> Luettu: 25.4.2018

Spring.io 2014. Spring Boot released. Luettavissa: <https://spring.io/blog/2014/01/22/spring-boot-1-0-0-rc1-released>. Luettu 14.10.2016.

Springtutorials 2015. Spring Framework history: 2002 - present. Luettavissa: <http://spring-tutorials.com/spring-framework-history/>. Luettu: 14.10.2016.

Teerimäki, J. 2012. IT-tradenomikoulutuksen vastaavuus työelämän vaatimuksiin. Amkopinnäytetyö. Haaga-Helia ammattikorkeakoulu. Helsinki. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/47952/ONT_Teerimaki_Joonas.pdf?sequence=1&isAllowed=y. Luettu: 23.4.2018

Toivanen, J.10.5.2018. Senior Specialist. Coheros. Haastattelu. Vantaa.

Virtanen, J. 3.1.2018. Osuiko Tieto Ruotsissa kultasuoneen? 12 viikon kurssit leipovat koodaajia kipeästi tarvittaviin töihin. Luettavissa: https://www.tivi.fi/Kaikki_uutiset/osuiko-tieto-ruotsissa-kultasuoneen-12-viikon-kurssit-leipovat-koodaajia-kipeasti-tarvittaviin-toihin-6694702. Luettu: 10.5.2018.

Walls, C. 2011. Spring in action, 3rd edition. Manning.

Walls, C. 2015. Spring boot in action. Manning.

Wellner, D. 22.06.2017. Java on kuollut, eläköön Kotlin! Luettavissa: <http://blogi.sysart.fi/java-on-kuollut-elakoon-kotlin>. Luettu: 25.4.2018

Widenius, M. 4.4.2018. Founder of MySQL AB. Luento, Tieto Boost for Techies Espoo, Tieto Keilalahti office.

Liitteet

Liite 1. Almutkan ja Puhakan tutkimuksen tulokset

	Ctg.	Topic	Learned in Educ.	Rank	Current Knowl.	Rank	Useful- ness in Work	Rank	Inf. on Think- ing	Rank	Profile
1	CSC	A Programming Language	3.2	2.	3.9	2.	4.0	2.	3.3	3.	D
2	CSC	Basic Programming Skills	3.5	1.	4.1	1.	4.1	1.	3.6	1.	D
3	CSC	Algorithm Design	2.4	15.	2.9	18.	2.9	22.	2.9	13.	D
4	CSC	Data Structures	3.1	3.	3.6	4.	3.5	10.	3.2	5.	D
5	CSC	Object-Oriented Programming	2.4	18.	3.1	14.	3.1	19.	2.9	11.	D
6	CSC	Programming Language Theory	2.4	17.	2.5	29.	2.2	35.	2.1	36.	
7	CSC	Parsing and Compiler Design	1.9	38.	2.2	41.	2.0	41.	1.8	49.	C
8	CSC	Databases	2.7	10.	3.7	3.	3.9	3.	3.4	2.	E
9	CSC	Operating Systems	2.7	11.	3.3	10.	3.2	17.	2.6	21.	D
10	CSC	Assembly Progr. and Embedded Syst.	1.6	54.	1.9	50.	1.9	46.	1.7	57.	C
11	CSC	Reliability and Fault Tolerance	1.7	47.	2.7	24.	3.1	18.	2.7	20.	E
12	CSC	Parallel and Distributed Computing	1.6	56.	1.9	51.	1.9	45.	1.8	48.	C
13	TCS	Performance Evaluation	1.9	36.	2.1	45.	2.1	39.	2.1	37.	C
14	TCS	Computability & Comput. Complexity	1.8	40.	1.7	57.	1.6	54.	1.8	51.	B
15	TCS	Automata Theory & Formal Lang.	1.8	43.	1.7	60.	1.6	55.	1.8	53.	B
16	TCS	Formal Program Verification	1.6	52.	1.6	62.	1.6	53.	1.6	60.	B
17	TCS	Formal Specification Methods	1.7	50.	1.8	53.	1.9	44.	1.9	47.	C
18	TCS	Information Theory	1.4	64.	1.5	67.	1.4	64.	1.5	62.	B
19	SoE	Requirements Gathering and Analysis	2.3	19.	3.4	7.	3.9	5.	3.3	4.	E
20	SoE	Specification and Design Methods	2.3	21.	2.9	20.	3.2	15.	2.8	16.	E
21	SoE	Software Architecture	2.1	25.	2.9	19.	3.0	20.	2.5	23.	E
22	SoE	Software Design and Patterns	2.0	34.	2.7	23.	2.9	24.	2.6	22.	E
23	SoE	Software Processes	2.2	23.	3.1	15.	3.3	13.	2.7	19.	E
24	SoE	Testing, Verification and Validation	2.1	28.	3.3	11.	3.7	8.	3.0	10.	E
25	SoE	Softw. Maintenance & Reengineering	2.0	32.	3.6	5.	3.7	6.	3.0	9.	E
26	SoE	Documentation	2.4	16.	3.5	6.	3.9	4.	3.1	6.	E
27	SoE	Config. & Release Management	1.5	62.	2.7	25.	2.9	23.	2.3	31.	E
28	SoE	Usability and User Interfaces	2.3	20.	3.2	12.	3.4	12.	3.0	8.	E
29	SoE	Software Metrics	1.3	70.	1.5	64.	1.6	57.	1.5	63.	
30	SoE	Project Management	2.1	24.	3.2	13.	3.5	11.	3.1	7.	E
31	SoE	Software Cost Estimation	1.4	67.	2.4	34.	2.9	25.	2.4	27.	E
32	SoE	Software Process Standards	1.6	57.	1.9	49.	2.0	40.	1.8	50.	C
33	ApS	Software in Digital Networks	2.1	27.	2.7	22.	2.8	26.	2.4	30.	D
34	ApS	Internet and World Wide Web	2.1	26.	3.4	9.	3.2	16.	2.8	15.	
35	ApS	Other Hypermedia and Multimedia	1.8	42.	2.4	35.	2.1	38.	2.0	38.	C
36	ApS	Mobile and Wireless Software	1.3	69.	2.1	44.	2.2	36.	1.9	45.	E
37	ApS	Security and Cryptography	1.9	37.	2.7	26.	3.0	21.	2.7	18.	E
38	ApS	Computer Graphics	1.7	48.	2.2	38.	1.8	47.	1.9	46.	
39	ApS	Artificial Intelligence	1.6	55.	1.8	55.	1.5	61.	1.8	54.	B
40	ApS	Pattern Recogn. & Image Processing	1.4	66.	1.7	56.	1.5	60.	1.6	61.	B
41	ApS	Comput. Methods for Numerical Probl.	1.8	45.	1.8	52.	1.8	49.	1.7	58.	B
42	ApS	Simulation	1.7	51.	1.9	46.	2.0	43.	1.9	44.	C
43	OIT	Computer System Architecture	2.5	13.	2.9	16.	2.6	29.	2.4	29.	C
44	OIT	Microprocessors	2.1	29.	2.2	39.	1.8	48.	1.8	55.	
45	OIT	Microprocessor Design	1.3	71.	1.2	72.	1.1	72.	1.3	70.	B
46	OIT	Digital Systems	1.7	49.	1.7	59.	1.4	63.	1.5	64.	B
47	OIT	Analog Electronics	1.5	58.	1.5	68.	1.2	68.	1.3	69.	B
48	OIT	Digital Communication Technologies	2.0	33.	2.4	33.	2.3	33.	1.9	42.	C
49	OIT	Digital Signal Processing	1.5	61.	1.5	65.	1.3	65.	1.3	68.	B
50	OIT	Mechatronics, Robotics	1.1	72.	1.2	71.	1.2	71.	1.2	72.	B
51	OIT	Measurement Technology	1.5	63.	1.4	69.	1.3	66.	1.4	66.	B
52	Mat	Differential and Integral Calculus	2.9	6.	2.2	40.	1.5	59.	1.9	41.	A
53	Mat	Linear Algebra and Matrices	2.8	9.	2.1	43.	1.7	50.	1.9	43.	A
54	Mat	Probability and Statistics	2.9	5.	2.5	31.	2.0	42.	2.3	32.	A
55	Mat	Logics	2.8	8.	2.6	28.	2.7	28.	2.7	17.	
56	Mat	Sets, Functions and Relations	2.9	7.	2.6	27.	2.2	34.	2.4	28.	A
57	Mat	Combinatorics	1.8	44.	1.7	61.	1.6	51.	1.8	52.	B
58	Mat	Graph Theory	1.8	39.	1.7	58.	1.6	52.	1.7	56.	B
59	Mat	Queueing Theory	1.6	53.	1.6	63.	1.4	62.	1.5	65.	B
60	Mat	Laplace and Fourier Transforms	1.8	41.	1.5	66.	1.2	70.	1.4	67.	A
61	Mat	Control Theory	1.4	68.	1.3	70.	1.2	69.	1.2	71.	B
62	Oth	Physics	2.6	12.	2.3	36.	1.5	58.	2.2	33.	A
63	Oth	Chemistry	2.0	30.	1.8	54.	1.2	67.	1.6	59.	A
64	Oth	Philosophy	1.5	59.	1.9	48.	1.6	56.	2.0	39.	
65	Oth	Psychology	1.4	65.	1.9	47.	2.3	32.	2.2	34.	E
66	Oth	Business Economics	2.5	14.	2.5	30.	2.6	31.	2.4	26.	C
67	Oth	Marketing	2.0	31.	2.2	42.	2.2	37.	2.0	40.	C
68	Oth	Leadership	1.8	46.	2.4	32.	2.8	27.	2.4	25.	E
69	Oth	Giving Presentations and Negotiation	2.3	22.	2.9	17.	3.6	9.	2.9	14.	E
70	Oth	Technical Writing	1.9	35.	2.8	21.	3.3	14.	2.4	24.	E
71	Oth	Foreign Languages	3.0	4.	3.4	8.	3.7	7.	2.9	12.	E
72	Oth	Software-related Law	1.5	60.	2.3	37.	2.6	30.	2.2	35.	E

Table 1: The averages for all 72 topics and 4 questions, including the rank of each average among the values in the same column, and the profile of each topic

Liite 2. Tutkimuskysely.

Ohjelmistokehittäjän työkalut

Tämä kysely on osa Haaga-Helia ammattikorkeakoulun tietojenkäsittelyn opiskelijan opinnäytetyötä. Tuloksia voidaan käyttää soveltuvin osin koulutusohjelman sisällön arviointiin ja kehittämiseen. Yksittäisen yrityksen vastauksia käytetään vain edellämäinnittuun tarkoitukseen. Tietoja ei luovuteta eteenpäin.

Yrityksen taustatiedot

Tietoja ei luovuteta eteenpäin.

1. Yrityksen ikä vuosissa

- ☐ 0-3
☐ 4-5
☐ 6-10
☐ 11 tai enemmän

2. Yrityksen henkilöstömäärä

- ☐ 1-10
☐ 11-50
☐ 51-250
☐ 251-500
☐ 501-1500
☒ 1501 tai enemmän

3. Yrityksen maantieteellinen sijainti, kotikunta.

Mikäli yrityksellä on toimipaikkoja useilla paikkakunnilla, kirjoitathan suurimman toimipisteen tai ensimmäisenä perustetun toimipisteen paikkakunnan.

4. Yrityksen tuotteet ja palvelut

Toteuttaako ja myykö yritys pääsääntöisesti omia sovelluksia vai onko yrityksen toiminta tilaus- tai konsultaatiopalvelua.

- ☒ Pääasiassa omia tuotteita
☐ Vain omia tuotteita
☐ Pääasiassa asiakkaan tilaamia projekteja (+ konsultaatio)
☐ Vain asiakkaan tilaamia projekteja (+ konsultaatio)
☐ Tasapuolisesti omia tuotteita sekä asiakasprojekteja

5. Vastaajan työnimike (vapaaehtoinen)

Millä nimikkeellä vastaaja työskentelee yrityksessä? Esimerkiksi ohjelmistokehittäjä, ohjelmistoarkkitehti, ohjelmistosuunnittelija..

Seuraava -->

(Sivu 1 / 5)

Yrityksen työkalut ja osaaminen

6. Mitä työkaluja yritys käyttää?

Mitä työkaluja, teknologioita, metodeita, toimintamalleja yrityksessä käytetään? Errottelethan vastaukset pilkulla.

<-- Edellinen

Seuraava -->

(Sivu 2 / 5)

Yrityksen työkalut ja osaaminen

7. Yrityksen työkalut ja niiden tärkeys

		Erittäin tärkeä	On ehkä tulevaisuudessa tärkeä	Melko tärkeä	On ollut tärkeä, ei enää	Ei käytössä
Java		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C#		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C++		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cobol		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu ohjelmointikieli, mikä?	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spring		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spring Boot		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
React native		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Android native		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu ohjelmistokehys, mikä?	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tietokannat		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MySql		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oracle		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SQL Server		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
NoSql		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu tietokantaratkaisu, mikä?	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SCRUM		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kanban		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu agile, mikä?	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Docker		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gradle		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Maven		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
DataWarehouse-työkalut, mikä/mitkä?	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Jokin muu, mikä?

Voit vielä lisätä tähän monivalintakysymyksen jälkeen mieleesi tulleita työkaluja tai kommentoida edellistä kysymystä.

9. Yrityksen muu osaaminen ja sen tärkeys

	Erittäin tärkeää	Melko tärkeää	Tarpeellista osaamista, vaikka ei niin tärkeää	En koe tärkeäksi yritykselleni
Dokumentaatio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vaatimusmäärittely	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ohjelmistoarkkitehtuuri	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testaus ja laadunvarmistus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Käytettävyys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Käyttäjäkokemus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suorituskyvyn optimointi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ylläpito ja versionhallinta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektinhallintataidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kokous- ja neuvottelutaidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Esiintymistaidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Myyntitaidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Asiakaspalvelutaidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Koulutustaidot	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Englanti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu kieli, mikä?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Jokin muu, mikä?

Voit vielä lisätä tähän monivalintakysymyksen jälkeen mieleesi tulleita osaamisalueita tai kommentoida edellistä kysymystä.

<-- Edellinen

Seuraava -->

(Sivu 3 / 5)

Uudet teknologiat

11. Yrityksen panostus uusiin teknologioihin

	Käytössä tai yritys tuottaa sisältöä tähän	Kiinnostava, mutta ei käytössä tai tuotannossa	Ei kiinnostava toistaiseksi
Internet of Things	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Virtual Reality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Artificial Intelligence	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Massadata-analytiikka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pilvipalvelut	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cyber Security	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Android	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
iOS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu mobiiliratkaisu? <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mac	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Linux	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Windows	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Muu käyttöjärjestelmä? <input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. Jokin muu, mikä?

Voit vielä lisätä tähän monivalintakysymyksen jälkeen mieleesi tulleita hype-statusen alaisia teknologioita tai kommentoida edellistä kysymystä.

[<-- Edellinen](#) [Seuraava -->](#)

(Sivu 4 / 5)

Kiitos vastauksista!

13. Terveiset kyselyn tekijälle

Jos mieleesi tuli vielä jotain, mitä haluaisit kysyä tai sanoa kyselyn tekijälle voit kirjoittaa sen allaolevaan kenttään. Tai jos tiedät yrityksen, joka voisi haluta vastata kyselyyn, voit myös kirjoittaa yrityksen nimen ja mahdollisen kontaktihenkilön. Suuri kiitos avustasi!

14. Vastaaajan yhteystiedot

Jos sinuun saa olla yhteydessä mahdollisten lisäkysymysten tiimoilta, jätä vielä yhteystietosi. Tietoja ei luovuteta eteenpäin.

Yritys	<input type="text"/>
Etunimi	<input type="text"/>
Sukunimi	<input type="text"/>
Matkapuhelin	<input type="text"/>
Sähköposti	<input type="text"/>
Muu kontaktointitapa	<input type="text"/>
Mieluisin yhteydenottotapa: puhelin / sähköposti / muu, mikä?	<input type="text"/>

[<-- Edellinen](#) [Lähetä](#)

(Sivu 5 / 5)

Liite 3. Osaamistarpeiden lista avoimien vastauksien tuloksista

Nimike	Määrä	Nimike	Määrä
JavaScript	28	HTML	4
Scrum	25	HTML5	4
Java	22	React Native	4
Git	18	Spring Boot	4
React JS	17	Visual Stuido Code	4
Agile	14	C++	3
Jira	14	Clojure	3
Amazon Web Services (AWS)	13	DevOps	3
Python	12	Github	3
Kanban	11	JavaEE	3
Docker	10	JVM	3
NodeJS	10	Kotlin	3
C#	9	Redux	3
iOS / macOS	9	Ruby on Rails	3
PHP	9	SVN (subversion)	3
AngularJS	8	Typescript	3
Jenkins	8	WildFly	3
Linux	8	Xcode	3
MySQL	8	(S)CSS	2
Slack	8	C	2
Visual studio 2014	8	Confluence	2
Eclipse	7	Delphi	2
IntelliJ Idea	7	Drupal	2
Microsoft Azure	7	Elm	2
Trello	7	ES6	2
.Net	6	HipChat	2
Ansible	6	Java8	2
Lean	6	JBoss	2
MongoDB	6	JQuery	2
PostgreSQL	6	Kaizen	2
Spring	6	Lean Product Canvas	2
Android	5	Magneto	2
Google cloud Platform	5	NoSQL	2
Scala	5	Objective-C	2
SQL	5	Oracle	2
SQL server management studio 2014	5	Postgres	2
Swift	5	Progressive Web Apps	2
WordPress	5	Reactive Programming	2
Angular	4	Redis	2
CSS	4	RXJS	2

Nimike	Määrä
SAFe	2
Scrumban	2
SolidDB	2
Sprint	2
Tomcat	2
Unix	2
Vaadin	2
Vesiputous	2
Vim	2
VScode	2
Windows	2
.Net Core	1
(S)FTP	1
AIX	1
AkkaHTTP	1
Apache camel	1
Aqua data tools	1
Artifactory	1
Asana	1
Atom	1
Atlassian tuotepihe *	1
Babel	1
Backbone	1
Bagrant	1
Bamboo	1
Bash	1
Bootstrap	1
BrowserStack	1
Centos	1
Checklist	1
Chef	1
CircleCI	1
Clojure script	1
Cobol	1
Code	1
Cognos	1
Concrete 5	1
Continuous integration (Jenkins etc)	1
Cordova	1
CouchDB	1
Cowboy Coding	1
Crucible	1
CSS3	1
CubeMeters	1
CVS	1
DaPulse	1
DB2	1

Nimike	Määrä
Design driven development	1
Design sprint	1
Diango	1
DMS	1
DreamWeaver	1
DS3	1
DynamoDB	1
EC2	1
Elasticsearch	1
Elixir	1
EQuela	1
Erlang	1
Express	1
EzPublish	1
F#	1
FishEye	1
Functional Programming	1
Google drive	1
Google suit	1
Gradle	1
Grails	1
Groovy	1
Hapi	1
Harvest	1
Haskell	1
Heroqu	1
Hibernate	1
HPE vertica	1
HTML4	1
HTML6	1
Hyper-V	1
IAAS	1
IBM content manager	1
intercom	1
Invision	1
Ionic	1
JS Frameworks	1
JSF	1
JSON	1
JSP	1
KnockoutJS	1
Kolvi	1
Liferay	1
Manifest .ly	1
MariaDB	1
Maven	1
MsSql	1

Nimike	Määrä
MVC	1
MVP	1
Netrex	1
Nginx	1
NoOps	1
Notepad ++	1
Object Oriented Design / Programming	1
Office 365	1
Ohjelmointi	1
Open office	1
Oracle SQL developer	1
Outsystems	1
PAW	1
Perl	1
Play Framework (scala versio)	1
PostGis	1
PowerShell	1
Puppet	1
Pycharm	1
Qt Creator	1
Razor	1
RDS	1
Reagent	1
Realm	1
Reveal	1
RIFD	1
Ruby	1
S3	1
Satellite	1
Selenium	1
Serverless	1
Skype	1
Sling	1
Smartmet	1
Sonar	1
Sovellettu Scrum	1
SSH	1
Stories on board	1
sublime text	1
Sympathy	1
Symphony	1
T-SQL	1
Tableau	1
Tekstieditorit	1
Terraform	1
Test-driven development	1
Tietokone	1

Nimike	Määrä
Twig	1
Typedrive	1
UML	1
Unity	1
User Journey Mapping	1
UWP	1
Varnish	1
Versionhallinta	1
VPC	1
VueJS	1
Web	1
Websphere portal	1
Windows 8.1	1
WINSCP	1
Xamarin	1
Zend	1

* Atlassian tuoteperhe:
confluence, jira, sourcetree, bitbucket

Liite 4. Teknologiset ratkaisut: Yrityksen työkalut ja niiden tärkeys. Monivalintakysymyksien kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä.

	Erittäin tärkeä	On ehkä tulevaisuudessa tärkeä	Melko tärkeä	On ollut tärkeä, ei enää	Ei käytössä	Vastaukset yhteensä
Java	28	0	8	5	15	56
C#	13	1	7	5	27	53
C++	3	0	4	7	38	52
Cobol	1	0	2	2	47	52
Muu ohjelmointikieli, mikä?	37	1	0	0	2	40
JavaScript	15	1	-	-	-	16
Groovy	1	-	-	-	-	1
Delphi	3	-	-	-	-	3
Ruby	2	-	-	-	-	2
Swift	2	-	-	-	-	2
Objective-C	1	-	-	-	-	1
Python	6	-	-	-	-	6
Elixir	1	-	-	-	-	1
PHP	7	-	-	-	-	7
Ecmascript	1	-	-	-	-	1
X++	1	-	-	-	-	1
Kotlin	1	-	-	-	-	1
Haskell	1	-	-	-	-	1
Scala	1	-	-	-	-	1
es6	1	-	-	-	-	1
Clojure	2	-	-	-	-	2
Typescript	1	-	-	-	-	1
Spring	12	3	7	5	23	50
Spring Boot	12	3	9	3	23	50
React native	10	10	5	0	24	49
Android native	5	5	11	2	24	47
Muu ohjelmistokehitys, mikä?	17	1	1	0	6	25
Skriptikielet / JS frameworkit, kirjastot	2	-	-	-	-	2
React / ReactJS	4	-	-	-	-	4
AngularJS	3	-	-	-	-	3
Angular 2/4 / Angular 2	2	-	-	-	-	2
VueJS	1	-	-	-	-	1
Node / NodeJS	1	-	-	-	-	1
jQuery	1	-	-	-	-	1
Sparkjava	1	-	-	-	-	1
Unity	1	-	-	-	-	1
iOS native / iOS (Swift)	1	1	-	-	-	2
hybrid & crossplatform kehitys	1	-	-	-	-	1
Django	1	-	-	-	-	1
ASP .net	1	-	-	-	-	1
Zend Framework	2	-	-	-	-	2
OutSystems	1	-	-	-	-	1
Ruby on Rails	1	-	-	-	-	1
Haskell (kieli)	-	-	1	-	-	1
Express	1	-	-	-	-	1
Webpack	1	-	-	-	-	1
Tietokannat	43	0	5	0	1	49
MySQL	27	0	12	4	11	54
Oracle	7	0	8	3	34	52
SQL Server	11	2	11	4	22	50
NoSql	14	6	11	0	20	51
Muu tietokantaratkaisu, mikä?	23	0	0	1	10	34
PostgreSQL	14	-	-	-	-	14
SolidDB	3	-	-	1	-	4
MongoDB	1	-	-	-	-	1
Redis	1	-	-	-	-	1
Elasticsearch	1	-	-	-	-	1
Couchbase	1	-	-	-	-	1
MariaDB	2	-	-	-	-	2
HPE Vertica	1	-	-	-	-	1
Hadoop	1	-	-	-	-	1
SCRUM	24	4	16	4	7	55
Kanban	20	4	12	4	14	54
Muu agile, mikä?	5	1	2	1	15	24
Lean thinking	2	-	1	-	-	3
SAFE	1	-	-	-	-	1
DevOps	-	1	-	-	-	1
Scrumban	-	-	1	-	-	1
Docker	31	10	5	0	11	57
Gradle	12	4	8	0	26	50
Maven	17	1	9	6	22	55
DataWarehouse-työkalut, mikä/mitkä?	6	3	2	0	18	29
MS BI	1	-	-	-	-	1
Vargant	1	-	-	-	-	1
Azure	1	-	-	-	-	1
Tableau	1	-	-	-	-	1
PowerBI	-	1	-	-	-	1
Impromptu	-	1	-	-	-	1
ETL	-	1	-	-	-	1
	Erittäin tärkeä	On ehkä tulevaisuudessa tärkeä	Melko tärkeä	On ollut tärkeä, ei enää	Ei käytössä	Vastaukset yhteensä

**Liite 5. Menetit ja toimintamallit: Yrityksen muu osaaminen ja sen tärkeys. Moniva-
lintakysymyksien kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä.**

	Erittäin tärkeää	Melko tärkeää	Tarpeellista osaamista, vaikka ei niin tärkeää	En koe tärkeäksi yritykselleni	Yhteensä
Dokumentaatio	21	20	14	3	58
Vaatimusmäärittely	32	14	7	4	57
Ohjelmistoarkkitehtuuri	35	18	2	2	57
Testaus ja laadunvarmistus	37	13	6	1	57
Käytettävyys	40	12	4	0	56
Käyttäjäkokeumus	41	13	4	0	58
Suorituskyvyn optimointi	20	24	11	2	57
Ylläpito ja versionhallinta	35	15	7	1	58
Projektinhallintataidot	23	25	9	0	57
Kokous- ja neuvottelutaidot	18	21	14	4	57
Esiintymistaidot	15	21	15	5	56
Myyntitaidot	19	18	12	8	57
Asiakaspalvelutaidot	30	15	11	1	57
Koulutustaidot	11	15	23	7	56
Englanti	28	19	8	2	57
Muu kieli, mikä?	4	2	1	4	11
Kiina	1	-	-	-	1
Suomi	3	1	-	-	4
Ruotsi	-	1	1	-	2
Jokin muu, mikä?	10	-	-	-	-
Tiimityöskentely	6	-	-	-	-
Itseopiskelu	1	-	-	-	-
Designlähtöinen/vetoinen kehitys	1	-	-	-	-
Kommunikaatiotaidot	2	-	-	-	-
Empatia	1	-	-	-	-
Ohjelmistokehitys	1	-	-	-	-
Rekrytointitaidot	1	-	-	-	-
Laadunhallinta	1	-	-	-	-
Teknisen velan hallinta	1	-	-	-	-
Management velan hallinta	1	-	-	-	-
Muutoksenhallinta	1	-	-	-	-
Mukautumiskyky	1	-	-	-	-
Omien virheiden hyväksyntä	1	-	-	-	-
Erlaisuuden kunnioitus ja hyväksyntä	1	-	-	-	-

Liite 6. Uudet teknologiat: Yrityksen panostus uusiin teknologioihin. Monivalintakysymyksien kaikki tulokset, sisältäen muu, mikä -kohdat eriteltynä.

	Käytössä tai yritys tuottaa sisältöä tähän	Kiinnostava, mutta ei käytössä tai tuotannossa	Ei kiinnostava toistaiseksi	Yhteensä
Internet of Things	15	22	18	55
Virtual Reality	7	22	26	55
Artificial Intelligence	15	27	13	55
Massadata-analytiikka	26	22	8	56
Pilvipalvelut	46	7	2	55
Cyber Security	22	19	13	54
Android	30	14	12	56
iOS	29	13	14	56
Muu mobiiliratkaisu?	5	1	11	17
React native	2	-	-	2
Hybrid- ja crossplatform	2	-	-	2
Websovellukset mobiilissa	1	-	-	1
Mac	25	11	20	56
Linux	32	11	13	56
Windows	28	9	19	56
Muu käyttöjärjestelmä?	1	0	15	16
Web	1	-	-	1
Jokin muu, mikä?				
Data Science	1	-	-	1
Machine Learning	1	-	-	1
Ohjelmistorobotiikka	1	-	-	1
Prosessien automatisointi	1	-	-	1
PWA	1	-	-	1
Augmented Reality	2	-	-	2
DevOps	1	-	-	1
Printed Intelligence	1	-	-	1
5G	1	-	-	1
Blockchain	1	-	-	1
Deep Learning	1	-	-	1